

# Developmental CARP Solving with GP

Thomas Weise, Alexandre Devert, and Ke Tang

Nature Inspired Computation and Applications Laboratory  
University of Science and Technology of China (USTC)  
Hefei 230027, Anhui, China

[tweise@ustc.edu.cn](mailto:tweise@ustc.edu.cn) · <http://www.it-weise.de/>  
Paper: <sup>[1]</sup>, see also <sup>[2]</sup>

Genetic and Evolutionary Computation Conference 2012  
2012-07-10 at Philadelphia, PA, USA

# Contents

- 1 Introduction
- 2 CARP
- 3 Phenotypes & Objective
- 4 Developmental Approach
- 5 Experiments
- 6 Summary



- 1 Introduction
- 2 CARP
- 3 Phenotypes & Objective
- 4 Developmental Approach
- 5 Experiments
- 6 Summary

# Introduction

# Introduction

## Developmental CARP Solving with Genetic Programming

# Introduction

## Developmental CARP Solving with Genetic Programming

- Capacitated Arc Routing Problem<sup>[3]</sup>

# Introduction

## Developmental CARP Solving with Genetic Programming

- Capacitated Arc Routing Problem<sup>[3]</sup>:
  - logistics planning problem

# Introduction

## Developmental CARP Solving with Genetic Programming

- Capacitated Arc Routing Problem<sup>[3]</sup>:
  - logistics planning problem
  - vehicles need to visit certain streets which require some treatment<sup>[3]</sup>

# Introduction

## Developmental CARP Solving with Genetic Programming

- Capacitated Arc Routing Problem<sup>[3]</sup>:
  - logistics planning problem
  - vehicles need to visit certain streets which require some treatment<sup>[3]</sup>
  - Examples:
    - road gritting and salting<sup>[7, 8]</sup>
    - (Chinese) Postman Problem<sup>[9, 10]</sup>

# Introduction

## Developmental CARP Solving with Genetic Programming

- Capacitated Arc Routing Problem<sup>[3]</sup>
- GP<sup>[4–6]</sup> ...

# Introduction

## Developmental CARP Solving with Genetic Programming

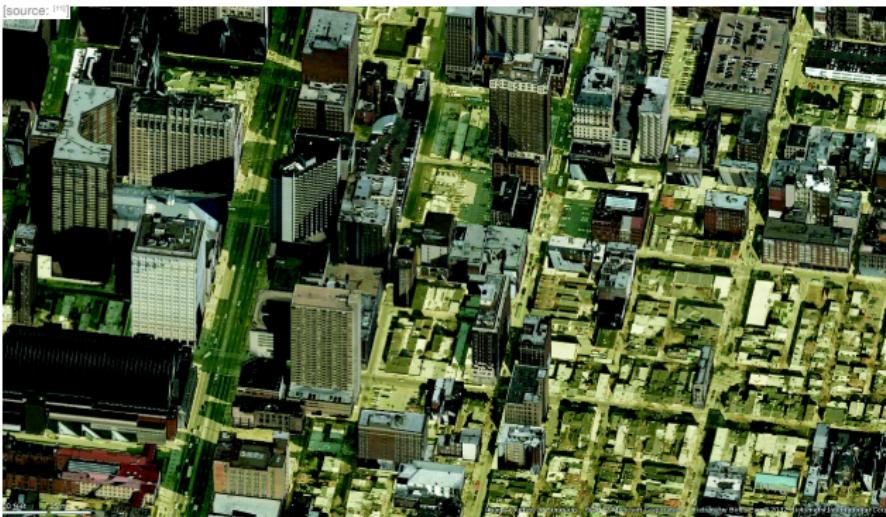
- Capacitated Arc Routing Problem [3]
- GP [4–6] ...
- Developmental Genotype-Phenotype Mapping:
  - build phenotype from genotype in an iterative process with feedback from environment

- 1 Introduction
- 2 CARP
- 3 Phenotypes & Objective
- 4 Developmental Approach
- 5 Experiments
- 6 Summary

# CARP

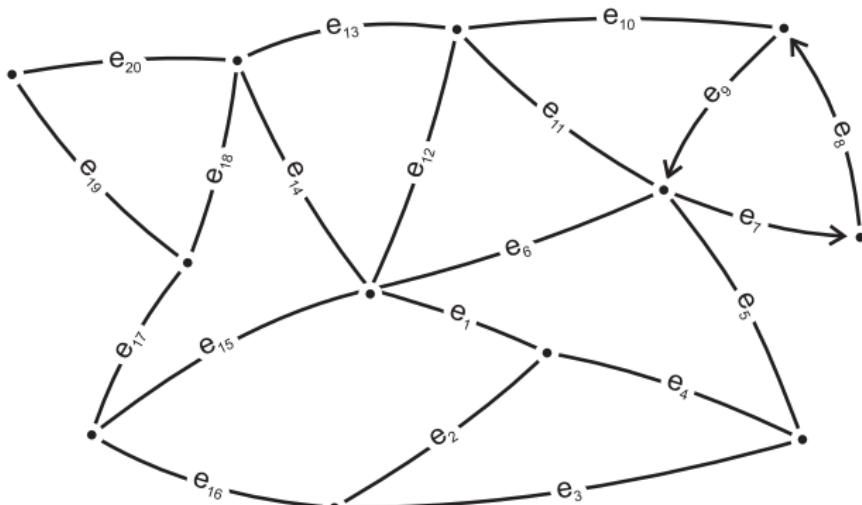
# Problem Definition

- Road network



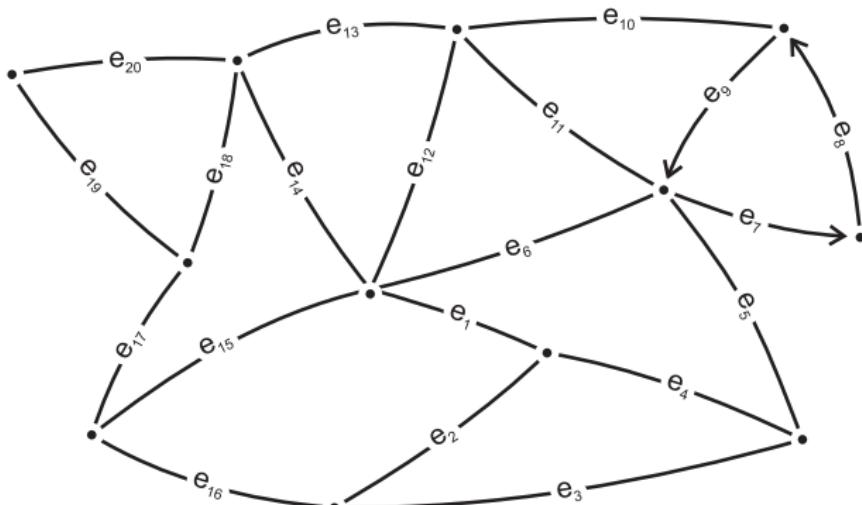
# Problem Definition

- Road network modeled as graph  $G = (V, E)$



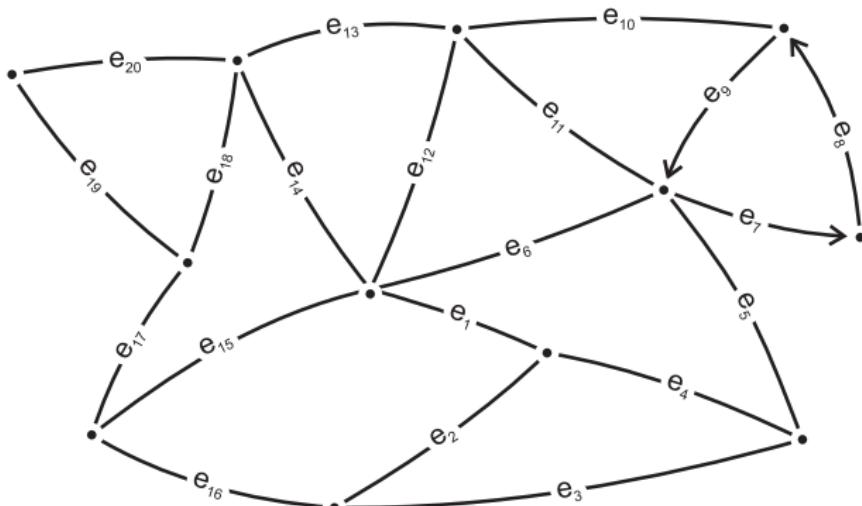
# Problem Definition

- Road network modeled as graph  $G = (V, E)$ 
  - a list  $E \subseteq V \times V$  of edges  $e$  (the streets) and



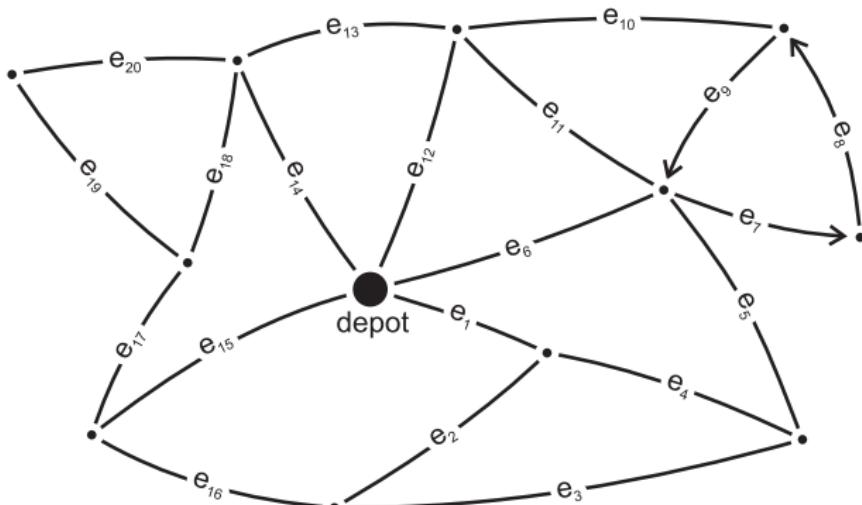
# Problem Definition

- Road network modeled as graph  $G = (V, E)$ 
  - a list  $E \subseteq V \times V$  of edges  $e$  (the streets) and
  - a set of vertices  $V$  (the intersections)



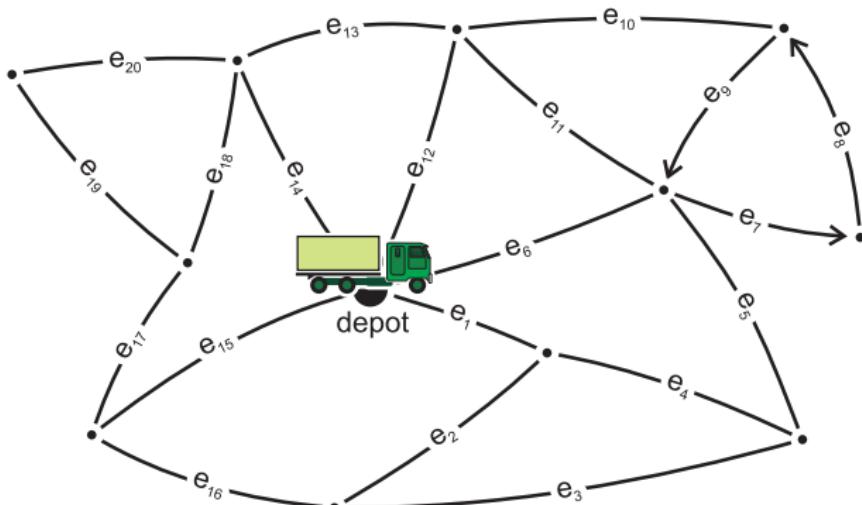
# Problem Definition

- The depot is a special vertex



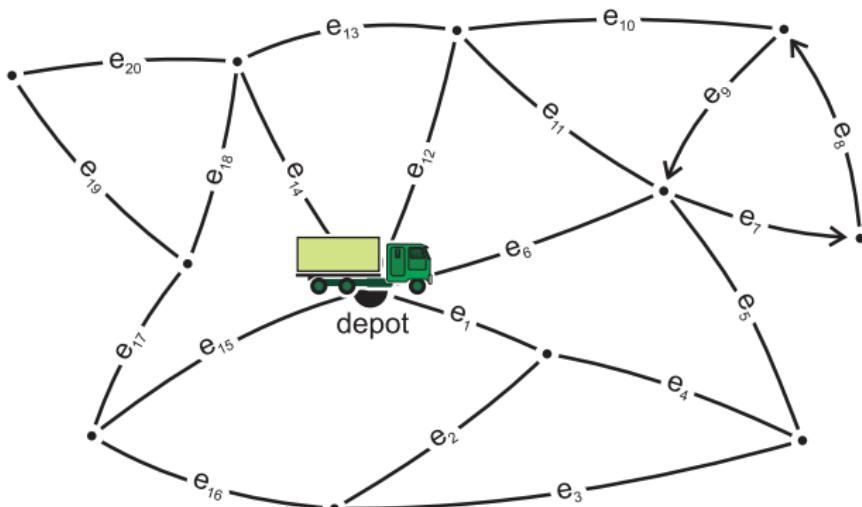
# Problem Definition

- The depot is a special vertex:
  - all vehicles are located at the depot and need to return at the end



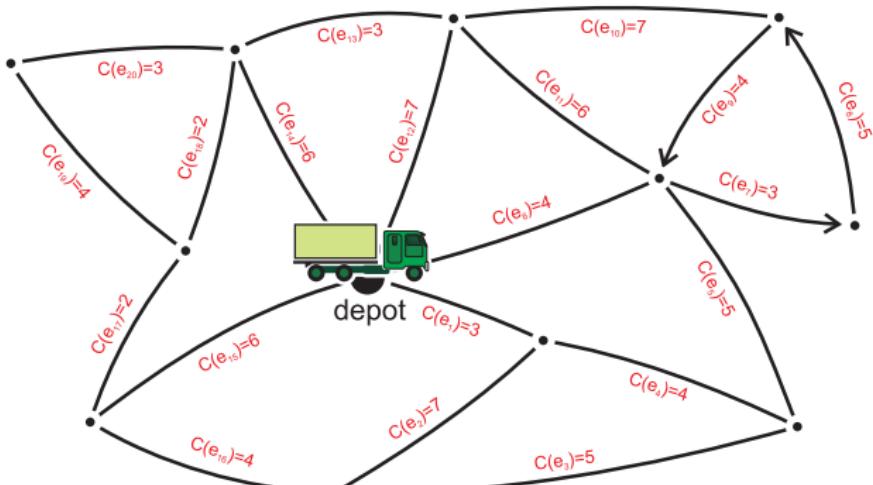
# Problem Definition

- The depot is a special vertex:
  - all vehicles are located at the depot and need to return at the end
  - the depot holds unlimited amounts of “the product”



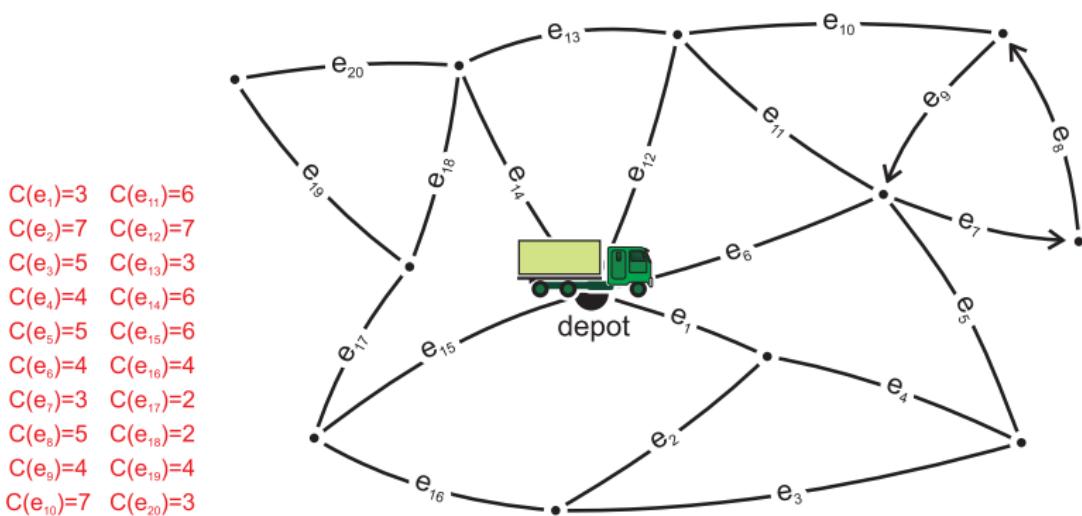
# Problem Definition

- Traversing an edge costs time/money/gasoline



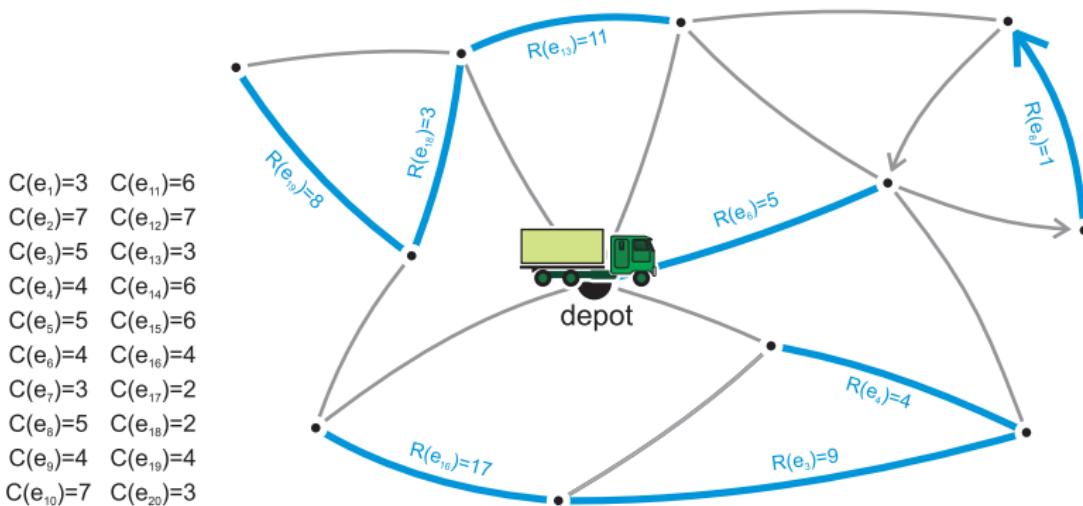
# Problem Definition

- Traversing an edge costs time/money/gasoline
  - cost function  $C : E \mapsto \mathbb{R}^+$



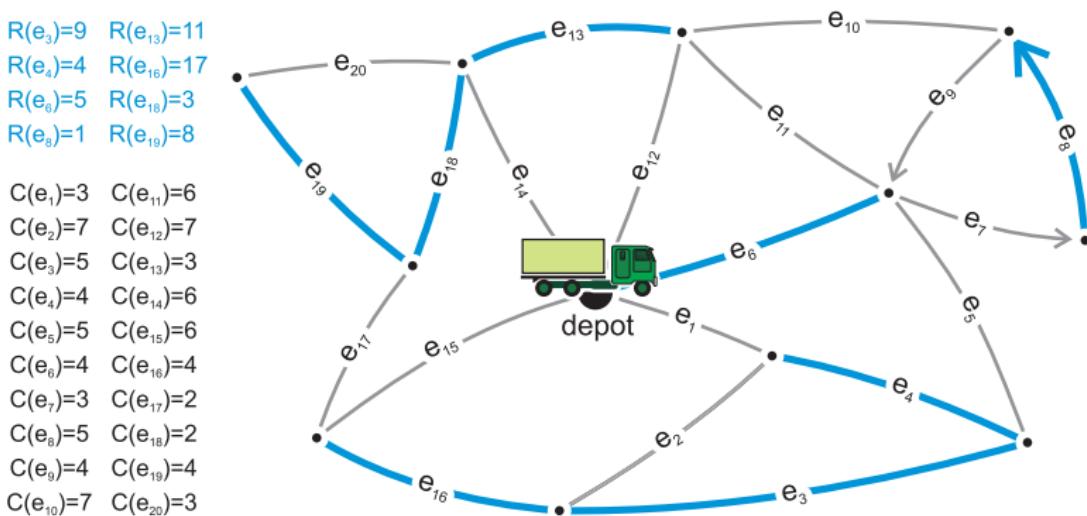
# Problem Definition

- Some edges have a demand for “the product”



# Problem Definition

- Some edges have a demand for “the product”
  - demand function  $R : E \mapsto \mathbb{N}_0$  (may be zero for some edges)



# Problem Definition

- Some edges have a demand for “the product”
  - demand function  $R : E \mapsto \mathbb{N}_0$  (may be zero for some edges)
  - (only) edges with non-zero demand *must* be visited

$$R(e_3)=9 \quad R(e_{13})=11$$

$$R(e_4)=4 \quad R(e_{16})=17$$

$$R(e_6)=5 \quad R(e_{18})=3$$

$$R(e_8)=1 \quad R(e_{19})=8$$

$$C(e_1)=3 \quad C(e_{11})=6$$

$$C(e_2)=7 \quad C(e_{12})=7$$

$$C(e_3)=5 \quad C(e_{13})=3$$

$$C(e_4)=4 \quad C(e_{14})=6$$

$$C(e_5)=5 \quad C(e_{15})=6$$

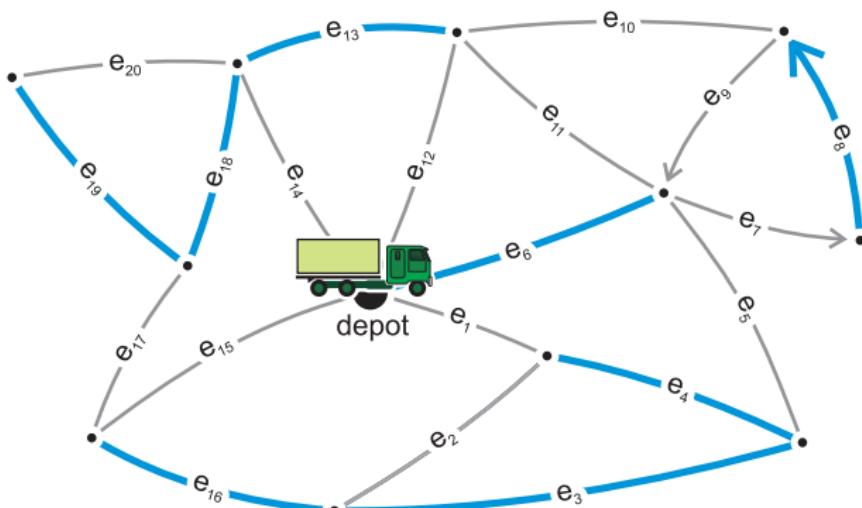
$$C(e_6)=4 \quad C(e_{16})=4$$

$$C(e_7)=3 \quad C(e_{17})=2$$

$$C(e_8)=5 \quad C(e_{18})=2$$

$$C(e_9)=4 \quad C(e_{19})=4$$

$$C(e_{10})=7 \quad C(e_{20})=3$$



# Problem Definition

- Some edges have a demand for “the product”
  - demand function  $R : E \mapsto \mathbb{N}_0$  (may be zero for some edges)
  - (only) edges with non-zero demand *must* be visited
  - each such edge’s demand must be satisfied in *exactly* one visit

$$R(e_3)=9 \quad R(e_{13})=11$$

$$R(e_4)=4 \quad R(e_{16})=17$$

$$R(e_6)=5 \quad R(e_{18})=3$$

$$R(e_8)=1 \quad R(e_{19})=8$$

$$C(e_1)=3 \quad C(e_{11})=6$$

$$C(e_2)=7 \quad C(e_{12})=7$$

$$C(e_3)=5 \quad C(e_{13})=3$$

$$C(e_4)=4 \quad C(e_{14})=6$$

$$C(e_5)=5 \quad C(e_{15})=6$$

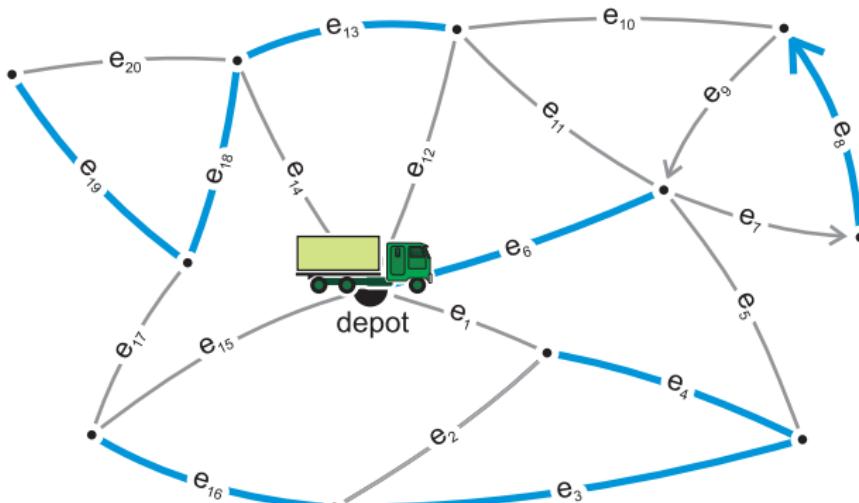
$$C(e_6)=4 \quad C(e_{16})=4$$

$$C(e_7)=3 \quad C(e_{17})=2$$

$$C(e_8)=5 \quad C(e_{18})=2$$

$$C(e_9)=4 \quad C(e_{19})=4$$

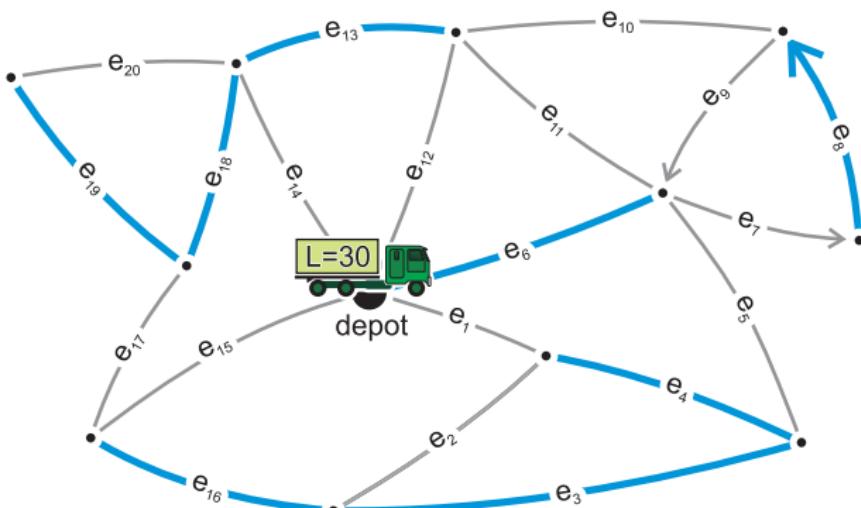
$$C(e_{10})=7 \quad C(e_{20})=3$$



# Problem Definition

- Vehicle(s) have capacity limit  $L \in \mathbb{N}_1$

$R(e_3)=9$	$R(e_{13})=11$
$R(e_4)=4$	$R(e_{16})=17$
$R(e_6)=5$	$R(e_{18})=3$
$R(e_8)=1$	$R(e_{19})=8$
$C(e_1)=3$	$C(e_{11})=6$
$C(e_2)=7$	$C(e_{12})=7$
$C(e_3)=5$	$C(e_{13})=3$
$C(e_4)=4$	$C(e_{14})=6$
$C(e_5)=5$	$C(e_{15})=6$
$C(e_6)=4$	$C(e_{16})=4$
$C(e_7)=3$	$C(e_{17})=2$
$C(e_8)=5$	$C(e_{18})=2$
$C(e_9)=4$	$C(e_{19})=4$
$C(e_{10})=7$	$C(e_{20})=3$

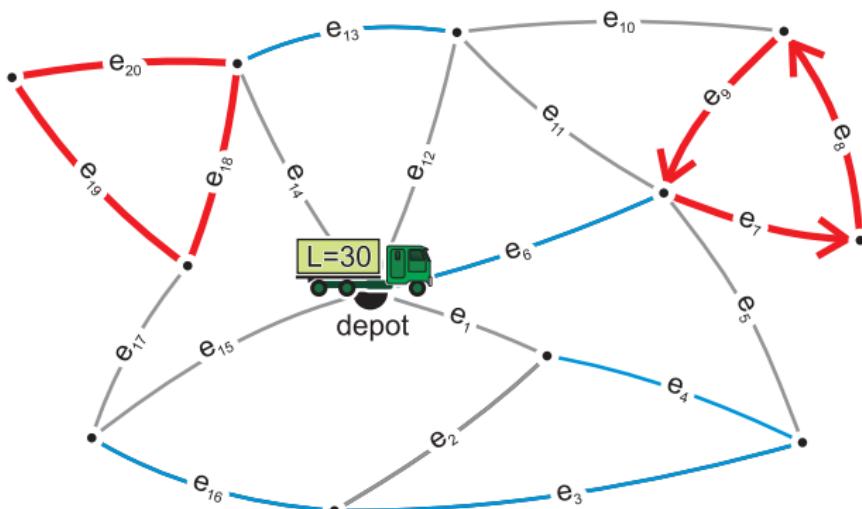


# Problem Definition

- Need solutions for graphs with directed and undirected edges

$R(e_3)=9 \quad R(e_{13})=11$   
 $R(e_4)=4 \quad R(e_{16})=17$   
 $R(e_6)=5 \quad R(e_{18})=3$   
 $R(e_8)=1 \quad R(e_{19})=8$

$C(e_1)=3 \quad C(e_{11})=6$   
 $C(e_2)=7 \quad C(e_{12})=7$   
 $C(e_3)=5 \quad C(e_{13})=3$   
 $C(e_4)=4 \quad C(e_{14})=6$   
 $C(e_5)=5 \quad C(e_{15})=6$   
 $C(e_6)=4 \quad C(e_{16})=4$   
 $C(e_7)=3 \quad C(e_{17})=2$   
 $C(e_8)=5 \quad C(e_{18})=2$   
 $C(e_9)=4 \quad C(e_{19})=4$   
 $C(e_{10})=7 \quad C(e_{20})=3$

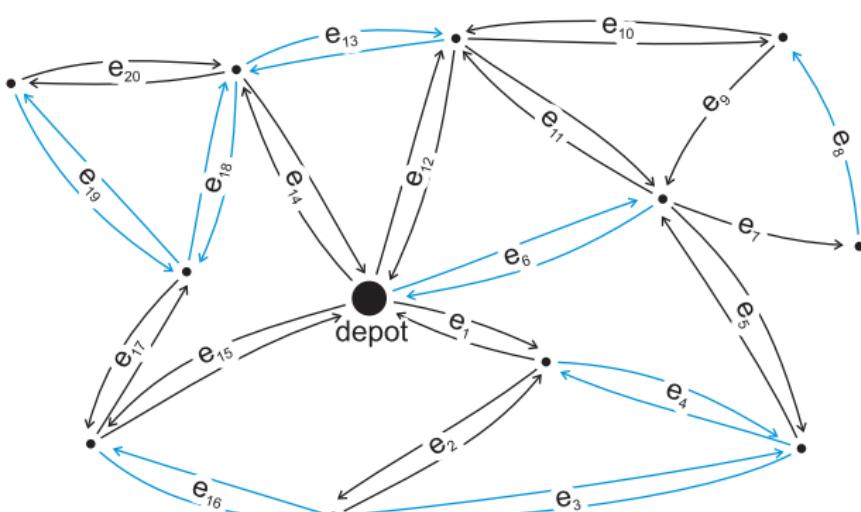


# Problem Definition

- Need solutions for graphs with directed and undirected edges:
  - represent undirected edges  $u = \overline{(v_i, v_j)}$  as two directed edges  $\overrightarrow{(v_i, v_j)}$  and  $\overleftarrow{(v_j, v_i)}$

$R(e_3)=9$     $R(e_{13})=11$   
 $R(e_4)=4$     $R(e_{16})=17$   
 $R(e_6)=5$     $R(e_{18})=3$   
 $R(e_8)=1$     $R(e_{19})=8$

$C(e_1)=3$     $C(e_{11})=6$   
 $C(e_2)=7$     $C(e_{12})=7$   
 $C(e_3)=5$     $C(e_{13})=3$   
 $C(e_4)=4$     $C(e_{14})=6$   
 $C(e_5)=5$     $C(e_{15})=6$   
 $C(e_6)=4$     $C(e_{16})=4$   
 $C(e_7)=3$     $C(e_{17})=2$   
 $C(e_8)=5$     $C(e_{18})=2$   
 $C(e_9)=4$     $C(e_{19})=4$   
 $C(e_{10})=7$     $C(e_{20})=3$



- 1 Introduction
- 2 CARP
- 3 Phenotypes & Objective
- 4 Developmental Approach
- 5 Experiments
- 6 Summary

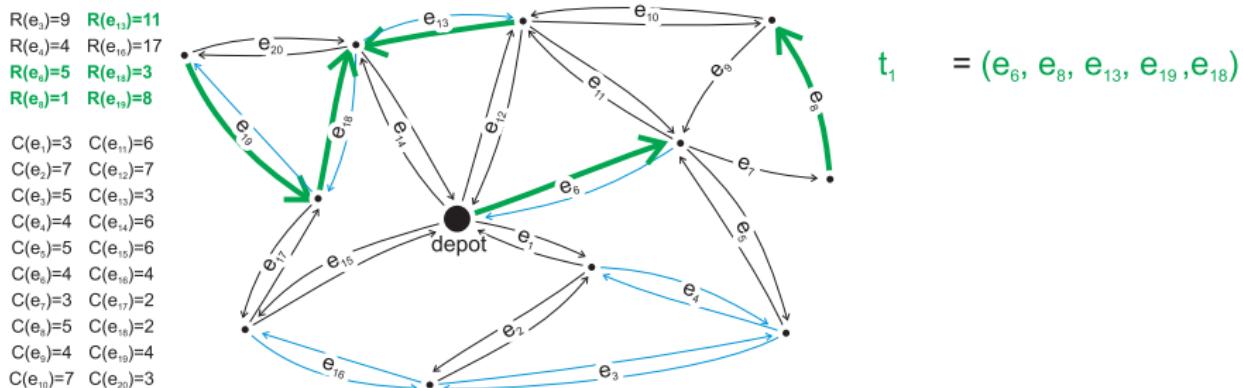
# Phenotypes & Objective

# Phenotypes & Objective

- Valid solution = satisfies all edge demands (with 1 visit/demand)

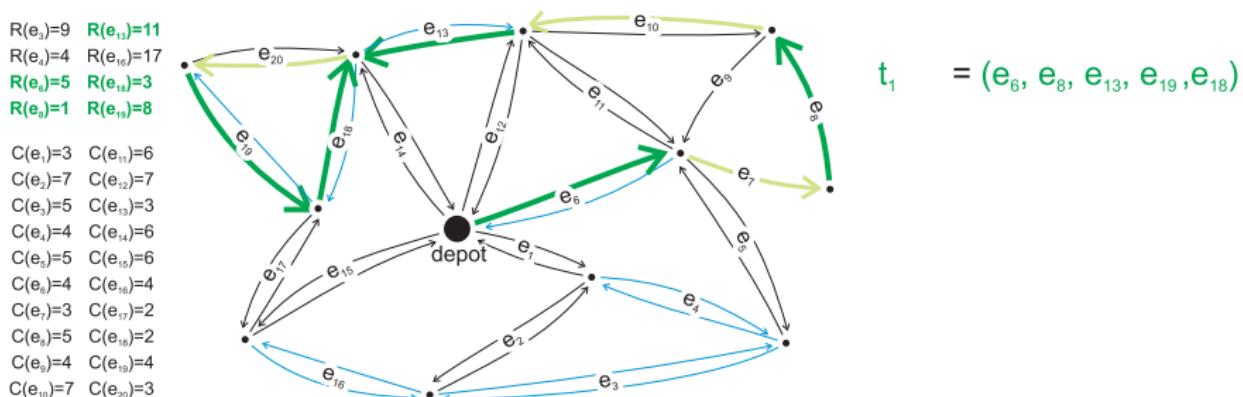
# Phenotypes & Objective

- Valid solution = satisfies all edge demands (with 1 visit/demand)
  - a tour  $t$  is a list of edges  $e \in E$  whose demand shall be satisfied



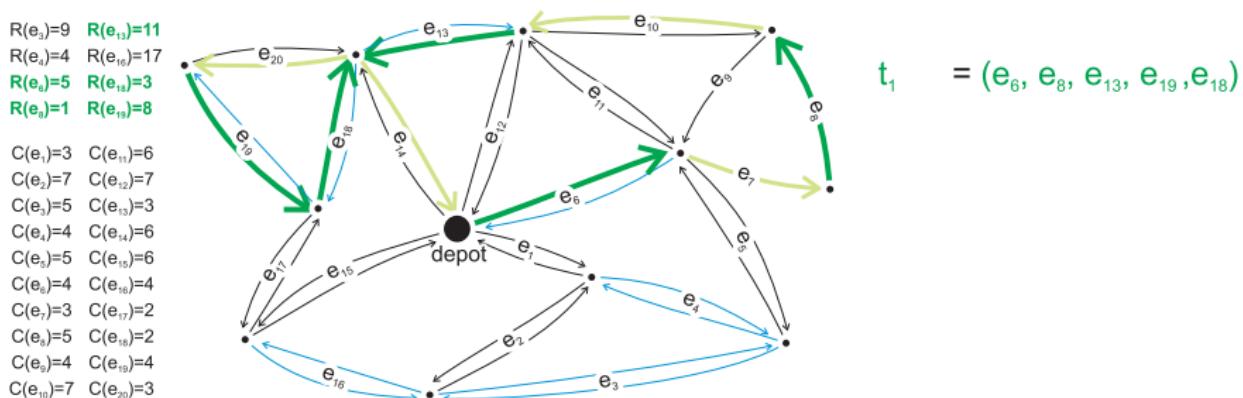
# Phenotypes & Objective

- Valid solution = satisfies all edge demands (with 1 visit/demand)
  - between these edges, minimum cost routing is assumed (using, e.g., Floyd's Algorithm<sup>[12]</sup>)



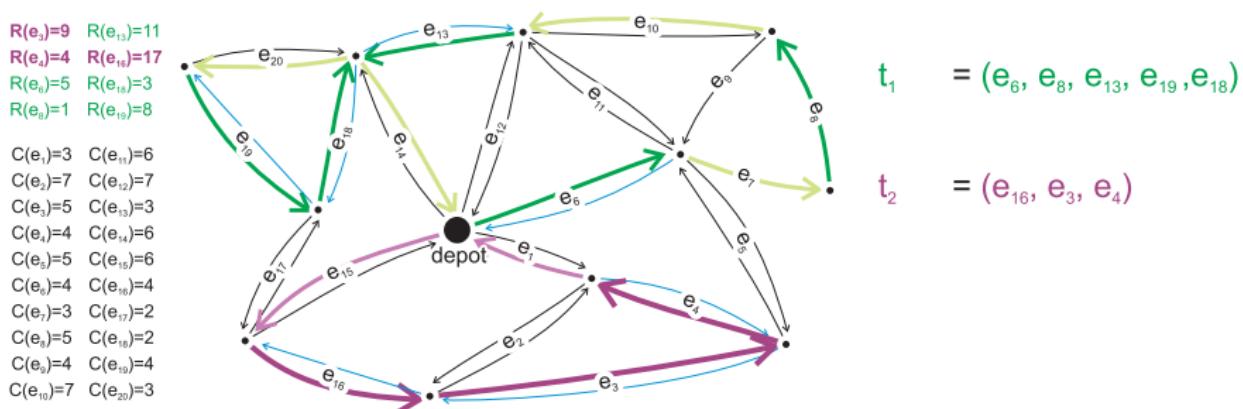
# Phenotypes & Objective

- Valid solution = satisfies all edge demands (with 1 visit/demand)
  - each tour starts and ends at the depot note  $v_1 \in V$



# Phenotypes & Objective

- Valid solution = satisfies all edge demands (with 1 visit/demand)
- $L < \sum_{\forall e \in E} \mathbf{R}(e) \implies$  multiple tours (or vehicles) needed

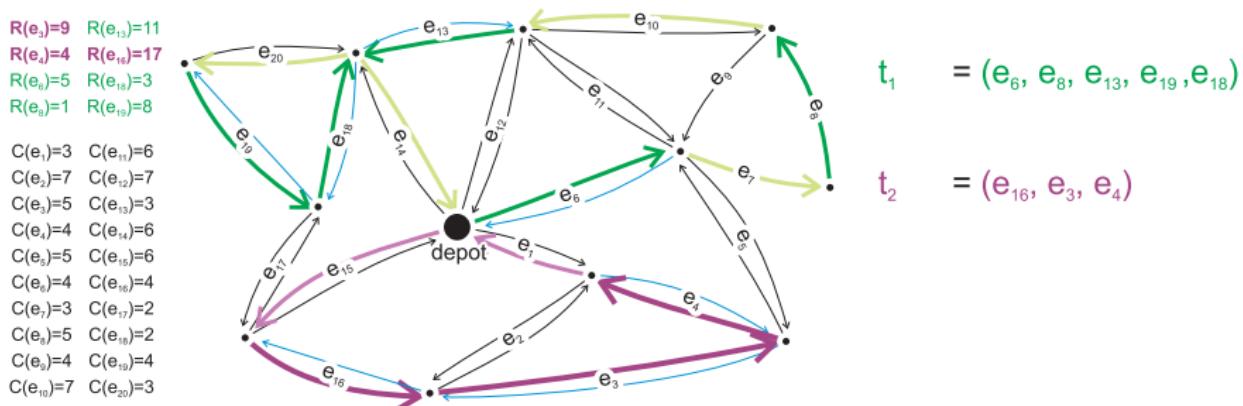


# Phenotypes & Objective

- Valid solution = satisfies all edge demands (with 1 visit/demand)

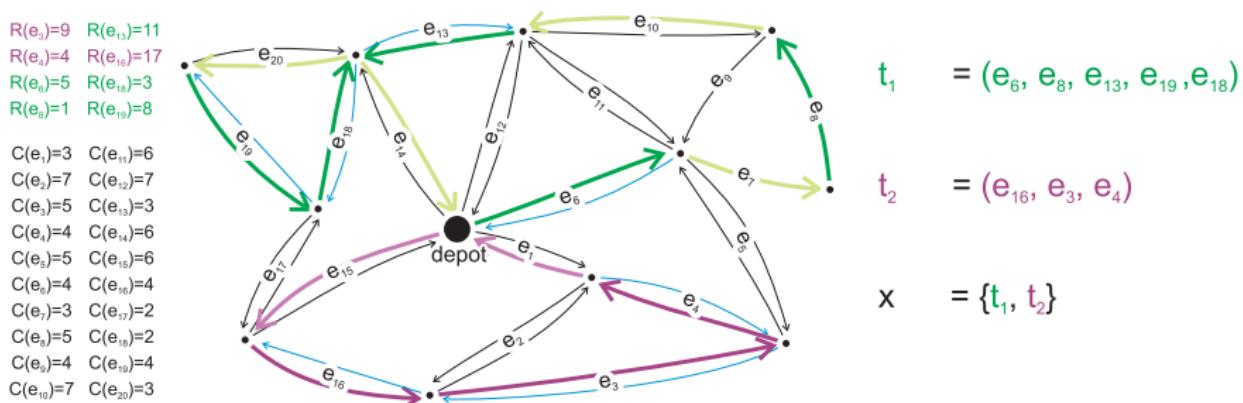
- $L < \sum_{\forall e \in E} \mathbf{R}(e) \implies$  multiple tours (or vehicles) needed

- of course  $L \geq \sum_{\forall e \in t} \mathbf{R}(e)$



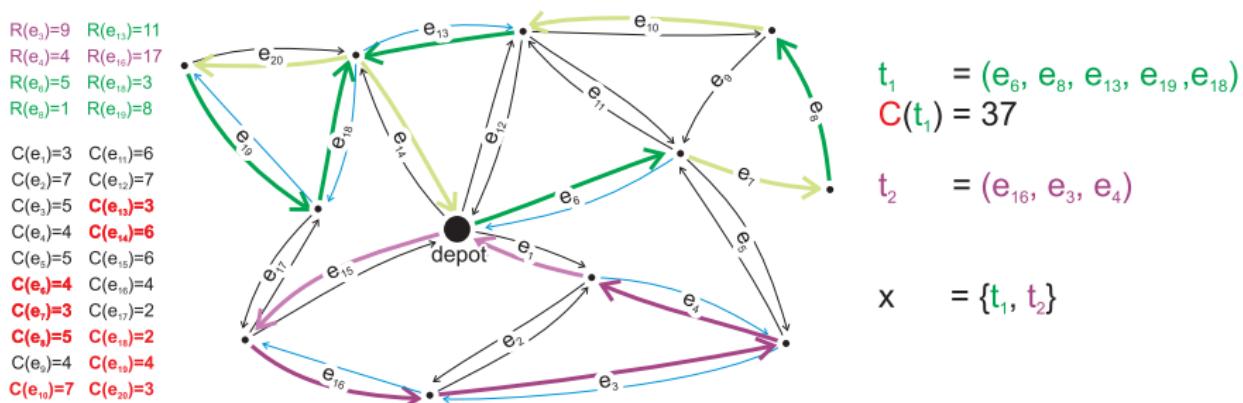
# Phenotypes & Objective

- Valid solution = satisfies all edge demands (with 1 visit/demand)
- Phenotype  $x \equiv$  set of tours that satisfies all demands (exactly once)



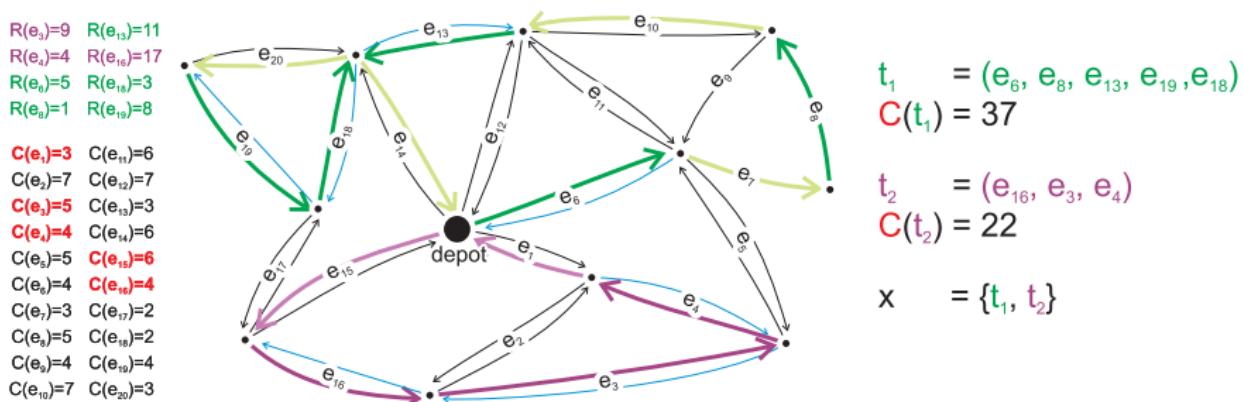
# Phenotypes & Objective

- Valid solution = satisfies all edge demands (with 1 visit/demand)
- Cost  $C(t)$  of tour  $t \equiv$  total traversal and routing costs



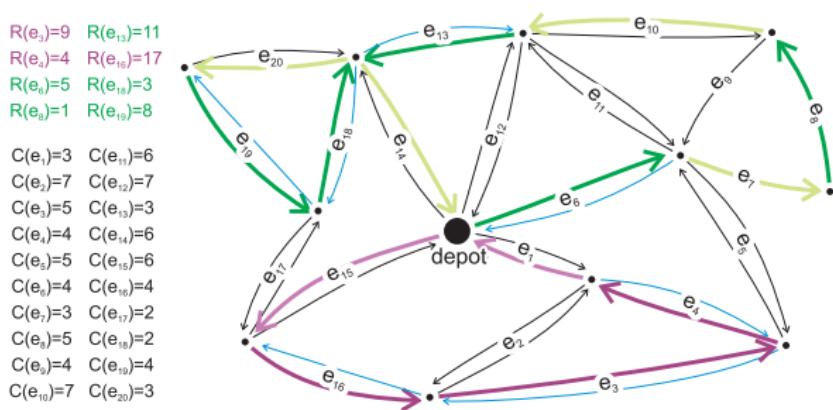
# Phenotypes & Objective

- Valid solution = satisfies all edge demands (with 1 visit/demand)
- Cost  $C(t)$  of tour  $t \equiv$  total traversal and routing costs



# Phenotypes & Objective

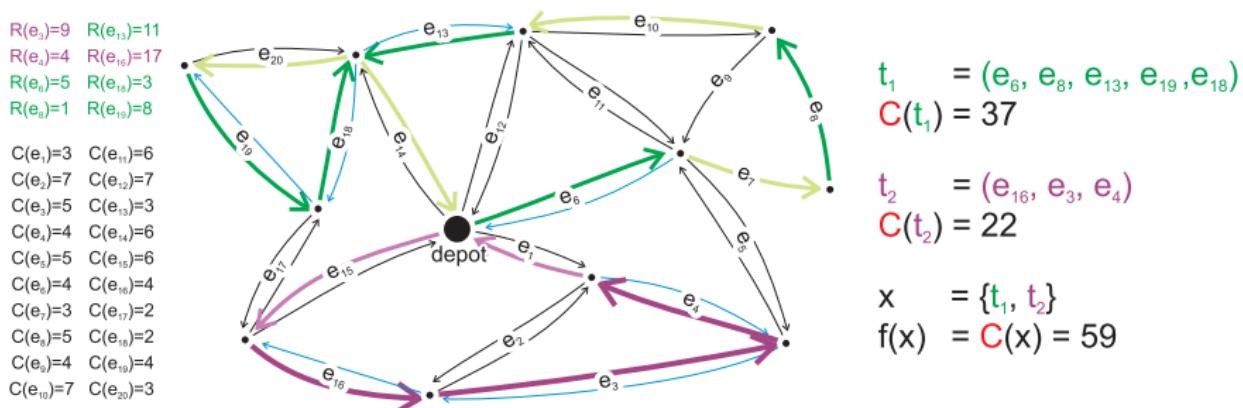
- Valid solution = satisfies all edge demands (with 1 visit/demand)
  - Cost  $\mathbf{C}(x)$  of solution  $x \equiv$  sum of all tour costs  $\left( \sum_{\forall t \in x} \mathbf{C}(t) \right)$



$$\begin{aligned}
 t_1 &= (e_6, e_8, e_{13}, e_{19}, e_{18}) \\
 C(t_1) &= 37 \\
 t_2 &= (e_{16}, e_3, e_4) \\
 C(t_2) &= 22 \\
 x &= \{t_1, t_2\} \\
 C(x) &= C(t_1) + C(t_2) = 59
 \end{aligned}$$

# Phenotypes & Objective

- Valid solution = satisfies all edge demands (with 1 visit/demand)
- Objective function  $f : \mathbb{X} \mapsto \mathbb{R}^+$  (subject to minimization):  $f(x) = \mathbf{C}(x)$



$$\begin{aligned}
 t_1 &= (e_6, e_8, e_{13}, e_{19}, e_{18}) \\
 \mathbf{C}(t_1) &= 37 \\
 t_2 &= (e_{16}, e_3, e_4) \\
 \mathbf{C}(t_2) &= 22 \\
 x &= \{t_1, t_2\} \\
 f(x) &= \mathbf{C}(x) = 59
 \end{aligned}$$

- 1 Introduction
- 2 CARP
- 3 Phenotypes & Objective
- 4 Developmental Approach
- 5 Experiments
- 6 Summary

# Developmental Approach

# Developmental Approach

- Genotypes are significantly different from phenotypes

# Developmental Approach

- Genotypes are significantly different from phenotypes
- A non-trivial genotype-phenotype mapping translates between them

# Developmental Approach

- Genotypes are significantly different from phenotypes
- A non-trivial genotype-phenotype mapping translates between them
- GPM uses information from

# Developmental Approach

- Genotypes are significantly different from phenotypes
- A non-trivial genotype-phenotype mapping translates between them
- GPM uses information from
  - ① the genotype

# Developmental Approach

- Genotypes are significantly different from phenotypes
- A non-trivial genotype-phenotype mapping translates between them
- GPM uses information from
  - ① the genotype
  - ② feedback from simulations or the process of computing the objective values

# Developmental Approach

- Genotypes are significantly different from phenotypes
- A non-trivial genotype-phenotype mapping translates between them
- GPM uses information from
  - ① the genotype
  - ② feedback from simulations or the process of computing the objective values
- GPM builds phenotypes in an iterative manner<sup>[13, 14]</sup>

# Developmental Approach

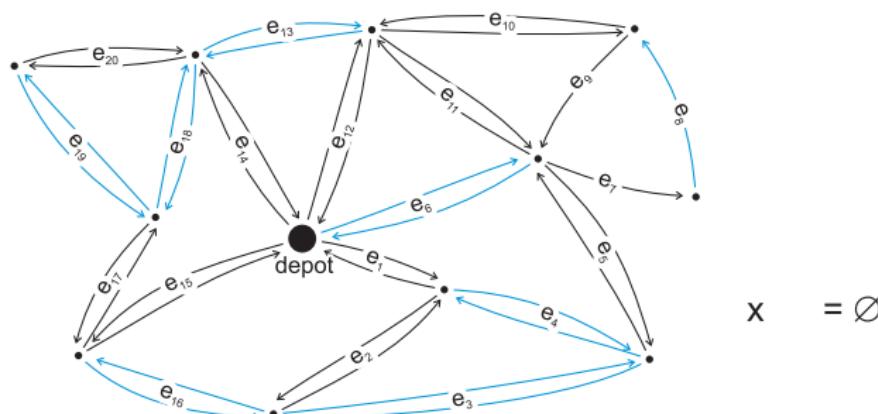
- Genotypes are significantly different from phenotypes
- A non-trivial genotype-phenotype mapping translates between them
- GPM uses information from
  - ① the genotype
  - ② feedback from simulations or the process of computing the objective values
- GPM builds phenotypes in an iterative manner<sup>[13, 14]</sup>
- Here: Evolve selection criterion (genotype) for **constructive heuristic** (GPM)

# Constructive Heuristic for CARP

- Constructive heuristic for a CARP

$R(e_1)=9 \quad R(e_{13})=11$   
 $R(e_2)=4 \quad R(e_{16})=17$   
 $R(e_3)=5 \quad R(e_{18})=3$   
 $R(e_4)=1 \quad R(e_{19})=8$

$C(e_1)=3 \quad C(e_{11})=6$   
 $C(e_2)=7 \quad C(e_{12})=7$   
 $C(e_3)=5 \quad C(e_{13})=3$   
 $C(e_4)=4 \quad C(e_{14})=6$   
 $C(e_5)=5 \quad C(e_{15})=6$   
 $C(e_6)=4 \quad C(e_{16})=4$   
 $C(e_7)=3 \quad C(e_{17})=2$   
 $C(e_8)=5 \quad C(e_{18})=2$   
 $C(e_9)=4 \quad C(e_{19})=4$   
 $C(e_{10})=7 \quad C(e_{20})=3$

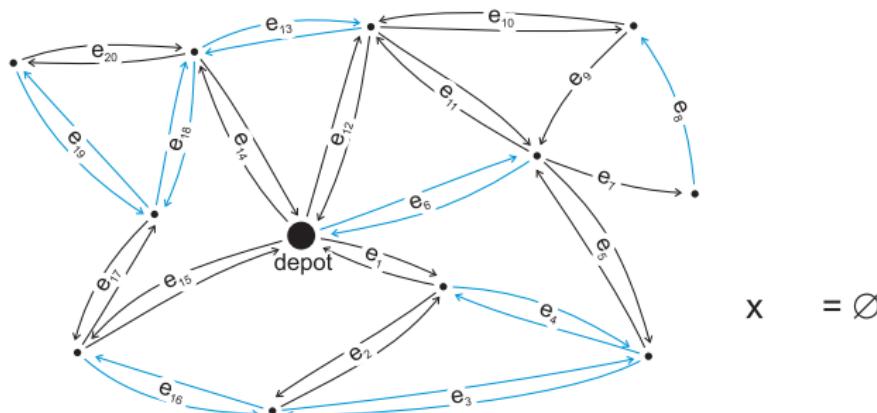


# Constructive Heuristic for CARP

- Constructive heuristic for a CARP
  - Start with empty schedule  $x$

$R(e_1)=9$     $R(e_{13})=11$   
 $R(e_2)=4$     $R(e_{16})=17$   
 $R(e_3)=5$     $R(e_{18})=3$   
 $R(e_4)=1$     $R(e_{19})=8$

$C(e_1)=3$     $C(e_{11})=6$   
 $C(e_2)=7$     $C(e_{12})=7$   
 $C(e_3)=5$     $C(e_{13})=3$   
 $C(e_4)=4$     $C(e_{14})=6$   
 $C(e_5)=5$     $C(e_{15})=6$   
 $C(e_6)=4$     $C(e_{16})=4$   
 $C(e_7)=3$     $C(e_{17})=2$   
 $C(e_8)=5$     $C(e_{18})=2$   
 $C(e_9)=4$     $C(e_{19})=4$   
 $C(e_{10})=7$     $C(e_{20})=3$

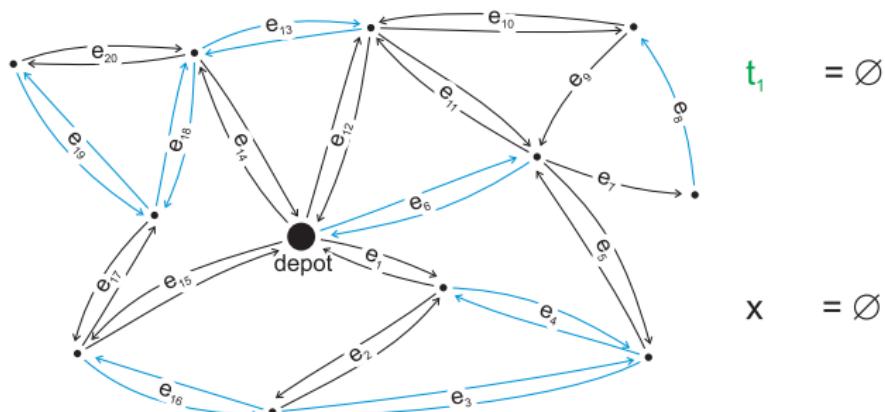


# Constructive Heuristic for CARP

- Constructive heuristic for a CARP
  - Iteratively add tours  $t$

$R(e_1)=9$     $R(e_{13})=11$   
 $R(e_2)=4$     $R(e_{16})=17$   
 $R(e_3)=5$     $R(e_{18})=3$   
 $R(e_4)=1$     $R(e_{19})=8$

$C(e_1)=3$     $C(e_{11})=6$   
 $C(e_2)=7$     $C(e_{12})=7$   
 $C(e_3)=5$     $C(e_{13})=3$   
 $C(e_4)=4$     $C(e_{14})=6$   
 $C(e_5)=5$     $C(e_{15})=6$   
 $C(e_6)=4$     $C(e_{16})=4$   
 $C(e_7)=3$     $C(e_{17})=2$   
 $C(e_8)=5$     $C(e_{18})=2$   
 $C(e_9)=4$     $C(e_{19})=4$   
 $C(e_{10})=7$     $C(e_{20})=3$

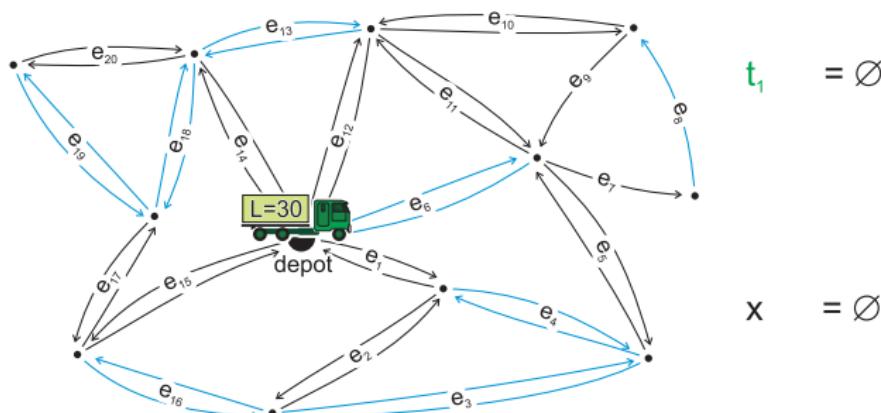


# Constructive Heuristic for CARP

- Constructive heuristic for a CARP
    - Iteratively add tours  $t$ :
- ① each tour starts at the depot  $v_1$

$R(e_1)=9$	$R(e_{13})=11$
$R(e_2)=4$	$R(e_{16})=17$
$R(e_3)=5$	$R(e_{18})=3$
$R(e_4)=1$	$R(e_{19})=8$

$C(e_1)=3$	$C(e_{11})=6$
$C(e_2)=7$	$C(e_{12})=7$
$C(e_3)=5$	$C(e_{13})=3$
$C(e_4)=4$	$C(e_{14})=6$
$C(e_5)=5$	$C(e_{15})=6$
$C(e_6)=4$	$C(e_{16})=4$
$C(e_7)=3$	$C(e_{17})=2$
$C(e_8)=5$	$C(e_{18})=2$
$C(e_9)=4$	$C(e_{19})=4$
$C(e_{10})=7$	$C(e_{20})=3$

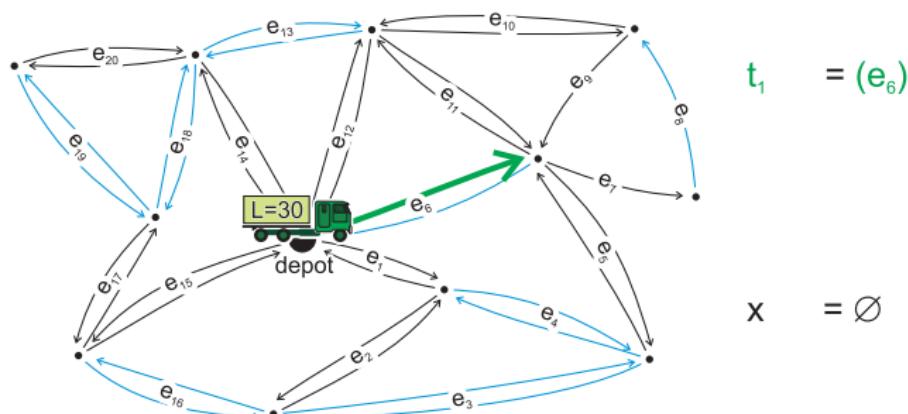


# Constructive Heuristic for CARP

- Constructive heuristic for a CARP
  - Iteratively add tours  $t$ :
  - ② pick an edge  $e^*$  from the set of unsatisfied edges

$R(e_3)=9$     $R(e_{13})=11$   
 $R(e_4)=4$     $R(e_{16})=17$   
 $R(e_5)=5$     $R(e_{18})=3$   
 $R(e_6)=1$     $R(e_{19})=8$

$C(e_1)=3$     $C(e_{11})=6$   
 $C(e_2)=7$     $C(e_{12})=7$   
 $C(e_3)=5$     $C(e_{13})=3$   
 $C(e_4)=4$     $C(e_{14})=6$   
 $C(e_5)=5$     $C(e_{15})=6$   
 $C(e_6)=4$     $C(e_{16})=4$   
 $C(e_7)=3$     $C(e_{17})=2$   
 $C(e_8)=5$     $C(e_{18})=2$   
 $C(e_9)=4$     $C(e_{19})=4$   
 $C(e_{10})=7$     $C(e_{20})=3$



# Constructive Heuristic for CARP

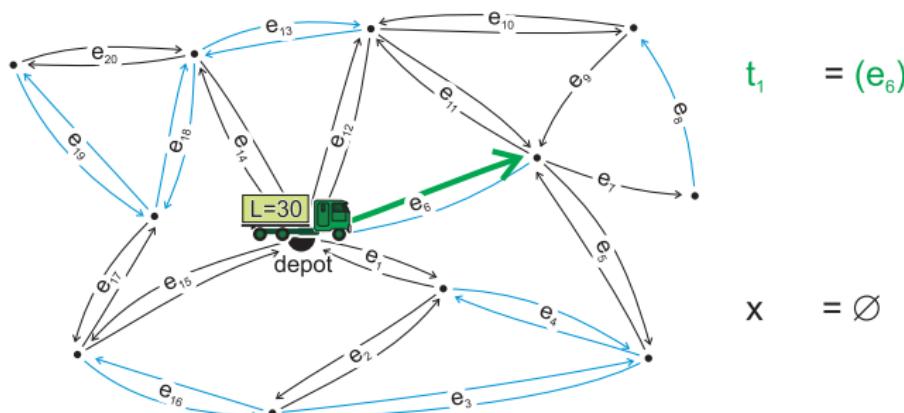
- Constructive heuristic for a CARP

- Iteratively add tours  $t$ :

- ② pick an edge  $e^*$  from the set of unsatisfied edges according to some criterion  $g$

$R(e_3)=9$     $R(e_{13})=11$   
 $R(e_4)=4$     $R(e_{16})=17$   
 $R(e_5)=5$     $R(e_{18})=3$   
 $R(e_6)=1$     $R(e_{19})=8$

$C(e_1)=3$     $C(e_{11})=6$   
 $C(e_2)=7$     $C(e_{12})=7$   
 $C(e_3)=5$     $C(e_{13})=3$   
 $C(e_4)=4$     $C(e_{14})=6$   
 $C(e_5)=5$     $C(e_{15})=6$   
 $C(e_6)=4$     $C(e_{16})=4$   
 $C(e_7)=3$     $C(e_{17})=2$   
 $C(e_8)=5$     $C(e_{18})=2$   
 $C(e_9)=4$     $C(e_{19})=4$   
 $C(e_{10})=7$     $C(e_{20})=3$

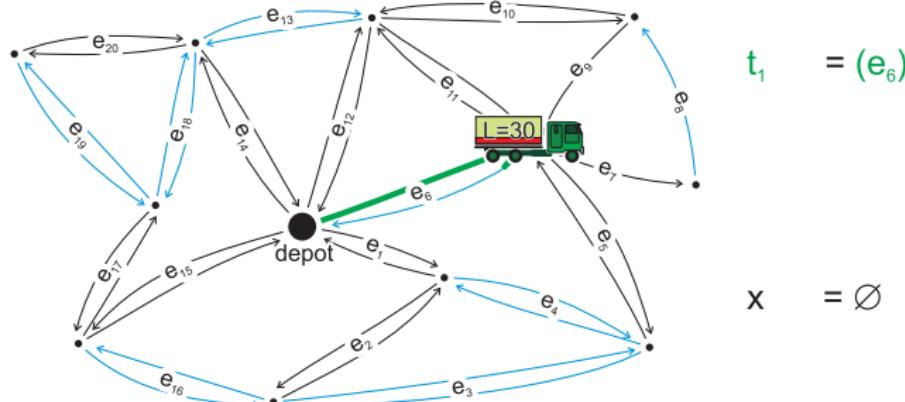


# Constructive Heuristic for CARP

- Constructive heuristic for a CARP
  - Iteratively add tours  $t$ :
  - ③ move vehicle to the end of  $e^*$  and decrease remaining capacity in vehicle

$R(e_3)=9$     $R(e_{13})=11$   
 $R(e_4)=4$     $R(e_{16})=17$   
 $R(e_5)=5$     $R(e_{18})=3$   
 $R(e_6)=1$     $R(e_{19})=8$

$C(e_1)=3$     $C(e_{11})=6$   
 $C(e_2)=7$     $C(e_{12})=7$   
 $C(e_3)=5$     $C(e_{13})=3$   
 $C(e_4)=4$     $C(e_{14})=6$   
 $C(e_5)=5$     $C(e_{15})=6$   
 $C(e_6)=4$     $C(e_{16})=4$   
 $C(e_7)=3$     $C(e_{17})=2$   
 $C(e_8)=5$     $C(e_{18})=2$   
 $C(e_9)=4$     $C(e_{19})=4$   
 $C(e_{10})=7$     $C(e_{20})=3$



# Constructive Heuristic for CARP

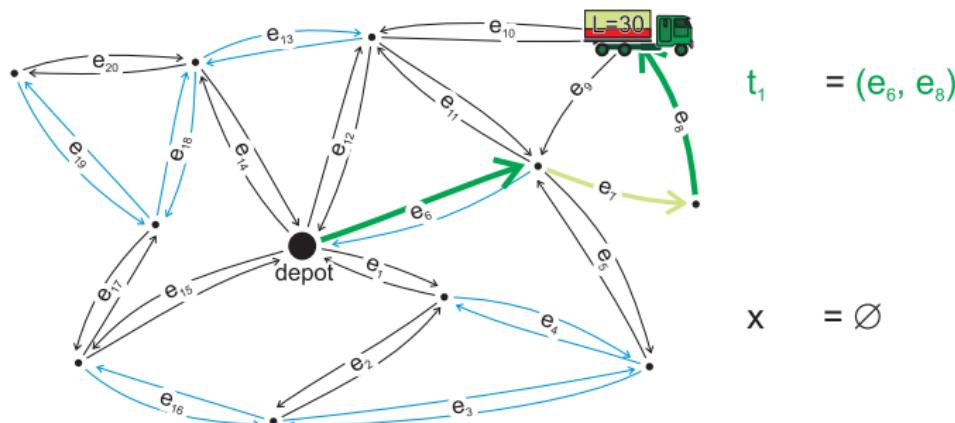
- Constructive heuristic for a CARP

- Iteratively add tours  $t$ :

④ repeat

$R(e_3)=9$	$R(e_{13})=11$
$R(e_4)=4$	$R(e_{16})=17$
$R(e_5)=5$	$R(e_{18})=3$
$R(e_6)=1$	$R(e_{19})=8$

$C(e_1)=3$	$C(e_{11})=6$
$C(e_2)=7$	$C(e_{12})=7$
$C(e_3)=5$	$C(e_{13})=3$
$C(e_4)=4$	$C(e_{14})=6$
$C(e_5)=5$	$C(e_{15})=6$
$C(e_6)=4$	$C(e_{16})=4$
$C(e_7)=3$	$C(e_{17})=2$
$C(e_8)=5$	$C(e_{18})=2$
$C(e_9)=4$	$C(e_{19})=4$
$C(e_{10})=7$	$C(e_{20})=3$



# Constructive Heuristic for CARP

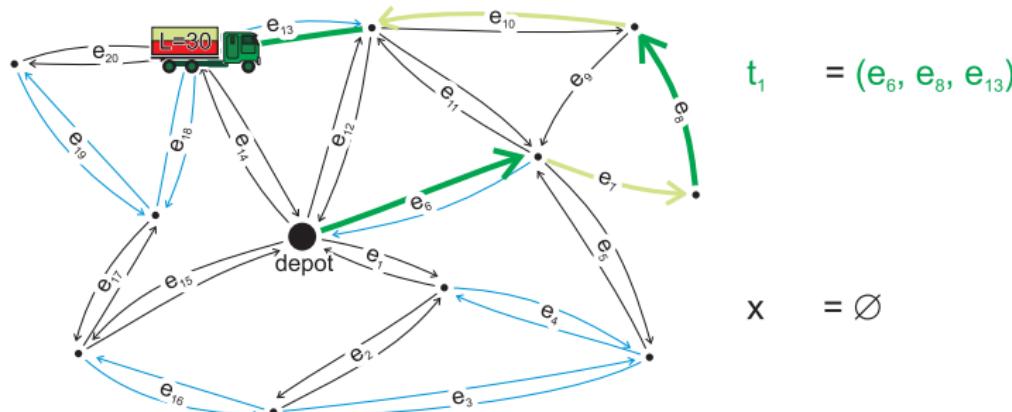
- Constructive heuristic for a CARP

- Iteratively add tours  $t$ :

④ repeat

$$\begin{array}{ll} R(e_1)=9 & R(e_{13})=11 \\ R(e_2)=4 & R(e_{16})=17 \\ R(e_3)=5 & R(e_{18})=3 \\ R(e_4)=1 & R(e_{19})=8 \end{array}$$

$$\begin{array}{ll} C(e_1)=3 & C(e_{11})=6 \\ C(e_2)=7 & C(e_{12})=7 \\ C(e_3)=5 & C(e_{13})=3 \\ C(e_4)=4 & C(e_{14})=6 \\ C(e_5)=5 & C(e_{15})=6 \\ C(e_6)=4 & C(e_{16})=4 \\ C(e_7)=3 & C(e_{17})=2 \\ C(e_8)=5 & C(e_{18})=2 \\ C(e_9)=4 & C(e_{19})=4 \\ C(e_{10})=7 & C(e_{20})=3 \end{array}$$



# Constructive Heuristic for CARP

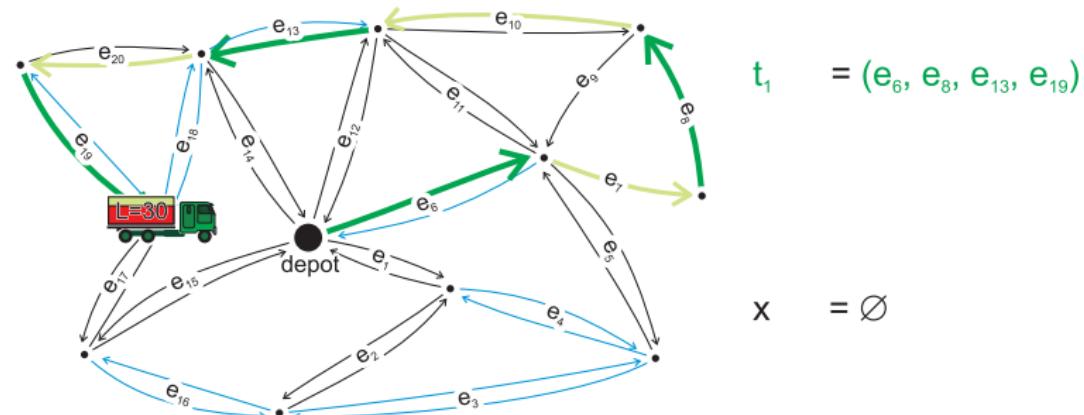
- Constructive heuristic for a CARP

- Iteratively add tours  $t$ :

④ repeat

$R(e_1)=9$	$R(e_{13})=11$
$R(e_2)=4$	$R(e_{16})=17$
$R(e_3)=5$	$R(e_{18})=3$
$R(e_4)=1$	$R(e_{19})=8$

$C(e_1)=3$	$C(e_{11})=6$
$C(e_2)=7$	$C(e_{12})=7$
$C(e_3)=5$	$C(e_{13})=3$
$C(e_4)=4$	$C(e_{14})=6$
$C(e_5)=5$	$C(e_{15})=6$
$C(e_6)=4$	$C(e_{16})=4$
$C(e_7)=3$	$C(e_{17})=2$
$C(e_8)=5$	$C(e_{18})=2$
$C(e_9)=4$	$C(e_{19})=4$
$C(e_{10})=7$	$C(e_{20})=3$



# Constructive Heuristic for CARP

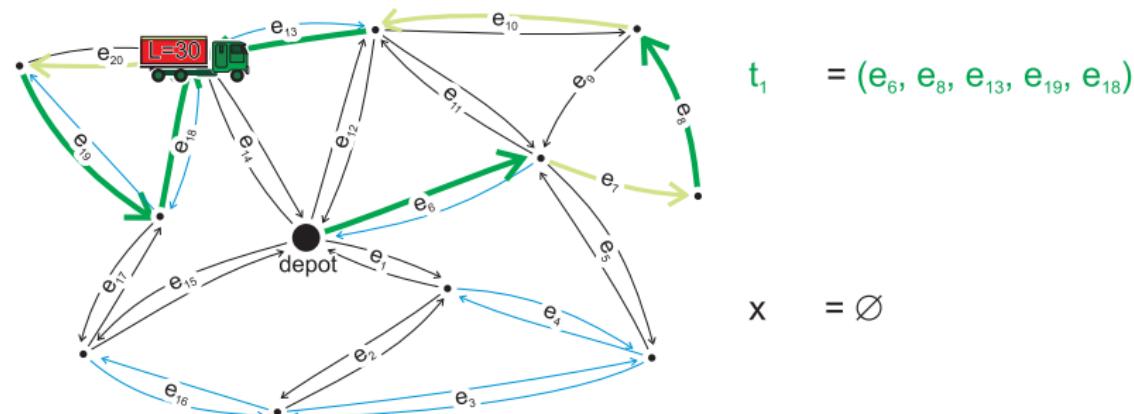
- Constructive heuristic for a CARP

- Iteratively add tours  $t$ :

④ repeat

$$\begin{array}{ll} R(e_1)=9 & R(e_{13})=11 \\ R(e_2)=4 & R(e_{14})=17 \\ R(e_3)=5 & R(e_{15})=3 \\ R(e_4)=1 & R(e_{16})=8 \end{array}$$

$$\begin{array}{ll} C(e_1)=3 & C(e_{11})=6 \\ C(e_2)=7 & C(e_{12})=7 \\ C(e_3)=5 & C(e_{13})=3 \\ C(e_4)=4 & C(e_{14})=6 \\ C(e_5)=5 & C(e_{15})=6 \\ C(e_6)=4 & C(e_{16})=4 \\ C(e_7)=3 & C(e_{17})=2 \\ C(e_8)=5 & C(e_{18})=2 \\ C(e_9)=4 & C(e_{19})=4 \\ C(e_{10})=7 & C(e_{20})=3 \end{array}$$



# Constructive Heuristic for CARP

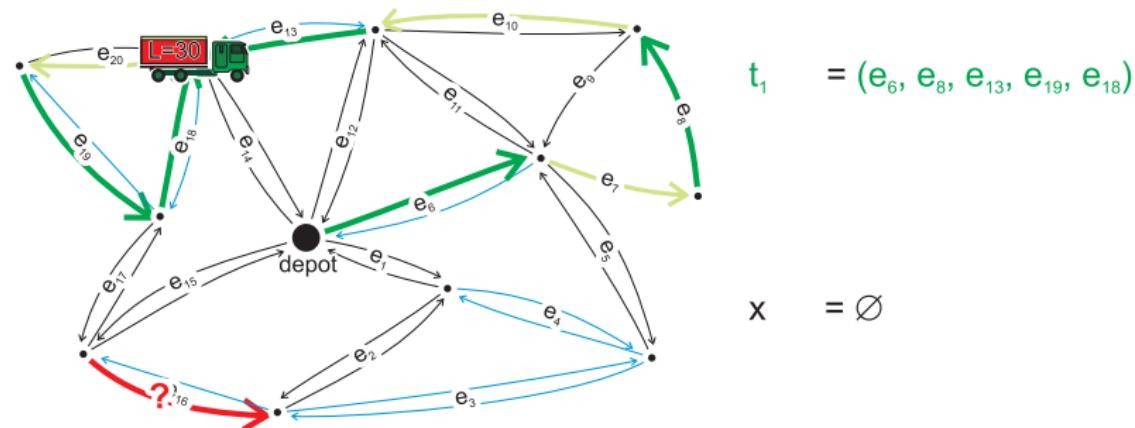
- Constructive heuristic for a CARP

- Iteratively add tours  $t$ :

- ④ repeat until chosen edge  $e^*$  cannot be satisfied because of capacity limit  $L$

$R(e_1)=9$     $R(e_{13})=11$   
 $R(e_2)=4$     $R(e_{14})=17$   
 $R(e_3)=5$     $R(e_{15})=3$   
 $R(e_4)=1$     $R(e_{16})=8$

$C(e_1)=3$     $C(e_{11})=6$   
 $C(e_2)=7$     $C(e_{12})=7$   
 $C(e_3)=5$     $C(e_{13})=3$   
 $C(e_4)=4$     $C(e_{14})=6$   
 $C(e_5)=5$     $C(e_{15})=6$   
 $C(e_6)=4$     $C(e_{16})=4$   
 $C(e_7)=3$     $C(e_{17})=2$   
 $C(e_8)=5$     $C(e_{18})=2$   
 $C(e_9)=4$     $C(e_{19})=4$   
 $C(e_{10})=7$     $C(e_{20})=3$

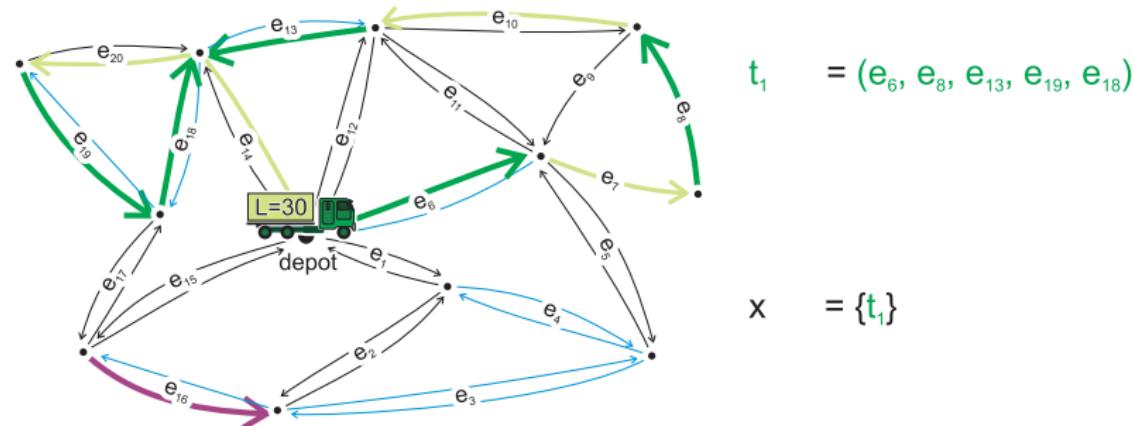


# Constructive Heuristic for CARP

- Constructive heuristic for a CARP
    - Iteratively add tours  $t$ :
- ⑤ add tour to phenotype and place vehicle back to depot

$R(e_1)=9$     $R(e_{13})=11$   
 $R(e_2)=4$     $R(e_{14})=17$   
 $R(e_3)=5$     $R(e_{15})=3$   
 $R(e_4)=1$     $R(e_{16})=8$

$C(e_1)=3$     $C(e_{11})=6$   
 $C(e_2)=7$     $C(e_{12})=7$   
 $C(e_3)=5$     $C(e_{13})=3$   
 $C(e_4)=4$     $C(e_{14})=6$   
 $C(e_5)=5$     $C(e_{15})=6$   
 $C(e_6)=4$     $C(e_{16})=4$   
 $C(e_7)=3$     $C(e_{17})=2$   
 $C(e_8)=5$     $C(e_{18})=2$   
 $C(e_9)=4$     $C(e_{19})=4$   
 $C(e_{10})=7$     $C(e_{20})=3$



# Constructive Heuristic for CARP

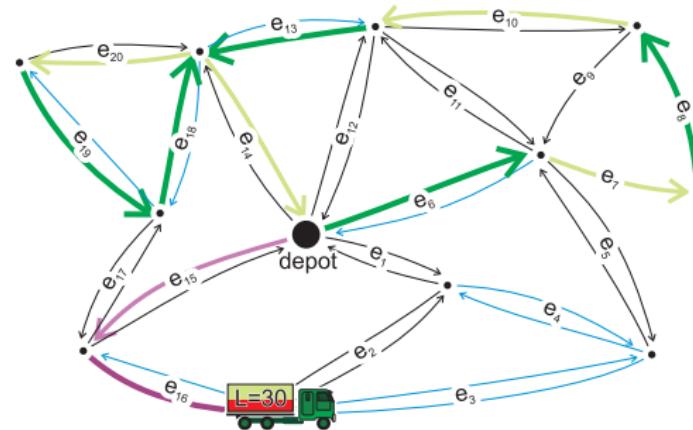
- Constructive heuristic for a CARP

- Iteratively add tours  $t$ :

- ⑥ continue with a new tour

$R(e_1)=9$     $R(e_{13})=11$   
 $R(e_2)=4$     $R(e_{14})=17$   
 $R(e_3)=5$     $R(e_{15})=3$   
 $R(e_4)=1$     $R(e_{16})=8$

$C(e_1)=3$     $C(e_{11})=6$   
 $C(e_2)=7$     $C(e_{12})=7$   
 $C(e_3)=5$     $C(e_{13})=3$   
 $C(e_4)=4$     $C(e_{14})=6$   
 $C(e_5)=5$     $C(e_{15})=6$   
 $C(e_6)=4$     $C(e_{16})=4$   
 $C(e_7)=3$     $C(e_{17})=2$   
 $C(e_8)=5$     $C(e_{18})=2$   
 $C(e_9)=4$     $C(e_{19})=4$   
 $C(e_{10})=7$     $C(e_{20})=3$



$$t_1 = (e_6, e_8, e_{13}, e_{18}, e_{19})$$

$$t_2 = (e_{16})$$

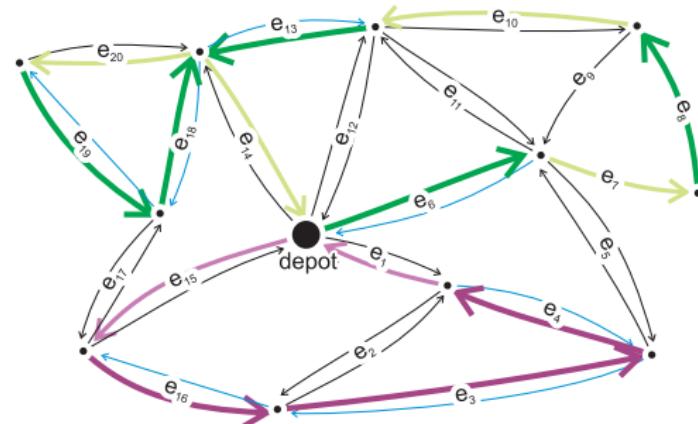
$$x = \{t_1\}$$

# Constructive Heuristic for CARP

- Constructive heuristic for a CARP
  - Iteratively add tours  $t$ :
  - ⑥ as long as there are unsatisfied demands, continue with a new tour

$R(e_1)=9$     $R(e_{13})=11$   
 $R(e_2)=4$     $R(e_{14})=17$   
 $R(e_3)=5$     $R(e_{15})=3$   
 $R(e_4)=1$     $R(e_{16})=8$

$C(e_1)=3$     $C(e_{11})=6$   
 $C(e_2)=7$     $C(e_{12})=7$   
 $C(e_3)=5$     $C(e_{13})=3$   
 $C(e_4)=4$     $C(e_{14})=6$   
 $C(e_5)=5$     $C(e_{15})=6$   
 $C(e_6)=4$     $C(e_{16})=4$   
 $C(e_7)=3$     $C(e_{17})=2$   
 $C(e_8)=5$     $C(e_{18})=2$   
 $C(e_9)=4$     $C(e_{19})=4$   
 $C(e_{10})=7$     $C(e_{20})=3$



$$\begin{aligned}
 t_1 &= (e_6, e_8, e_{13}, e_{18}, e_{19}) \\
 t_2 &= (e_{16}, e_3, e_4) \\
 X &= \{t_1, t_2\}
 \end{aligned}$$

# Constructive Heuristic for CARP

- Constructive heuristic for a CARP

- Start with empty schedule  $x$
- Iteratively add tours  $t$ :

- ① each tour starts at the depot  $v_1$
- ② pick an edge  $e^*$  from the set of unsatisfied edges according to some criterion  $g$
- ③ move vehicle to the end of  $e^*$  and decrease remaining capacity in vehicle
- ④ repeat until chosen edge  $e^*$  cannot be satisfied because of capacity limit  $L$
- ⑤ add tour to phenotype and place vehicle back to depot
- ⑥ as long as there are unsatisfied demands, continue with a new tour

# Constructive Heuristic for CARP

- Constructive heuristic for a CARP

- Start with empty schedule  $x$
- Iteratively add tours  $t$ :

- ① each tour starts at the depot  $v_1$
- ② pick an edge  $e^*$  from the set of unsatisfied edges ***according to some criterion g***
- ③ move vehicle to the end of  $e^*$  and decrease remaining capacity in vehicle
- ④ repeat until chosen edge  $e^*$  cannot be satisfied because of capacity limit  $L$
- ⑤ add tour to phenotype and place vehicle back to depot
- ⑥ as long as there are unsatisfied demands, continue with a new tour

# Developmental GPM

- Constructive heuristic is genotype-phenotype mapping

# Developmental GPM

- Constructive heuristic is genotype-phenotype mapping
- Use edge selection criterion  $g$  as genotypes

# Developmental GPM

- Constructive heuristic is genotype-phenotype mapping
- Use edge selection criterion  $g$  as genotypes
- Let  $g$  be a function  $g : E \mapsto \mathbb{R}$  assigning heuristic values edges

# Developmental GPM

- Constructive heuristic is genotype-phenotype mapping
- Use edge selection criterion  $g$  as genotypes
- Let  $g$  be a function  $g : E \mapsto \mathbb{R}$  assigning heuristic values edges
- Edge with smallest heuristic value is added

# Developmental GPM

- Constructive heuristic is genotype-phenotype mapping
- Use edge selection criterion  $g$  as genotypes
- Let  $g$  be a function  $g : E \cup \{\underline{e}\} \mapsto \mathbb{R}$  assigning heuristic values to edges
- Edge with smallest heuristic value is added
- Allow the heuristic to decide when to end a tour: artificial edge  $\underline{e} = \overrightarrow{(v_1, v_1)}$  representing visit to the depot

# Search Space

- How to encode such a heuristic function?

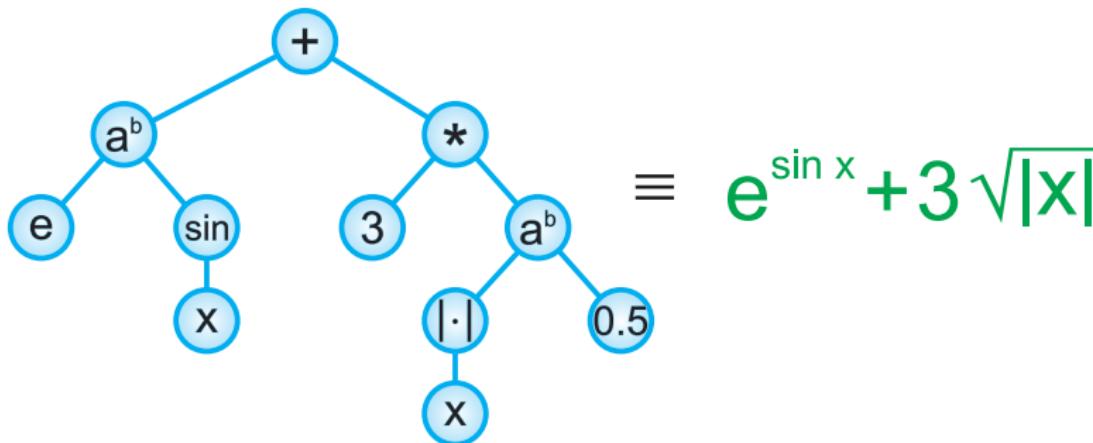
# Search Space

- How to encode such a heuristic function?
- Mathematical functions

$$e^{\sin x} + 3\sqrt{|x|}$$

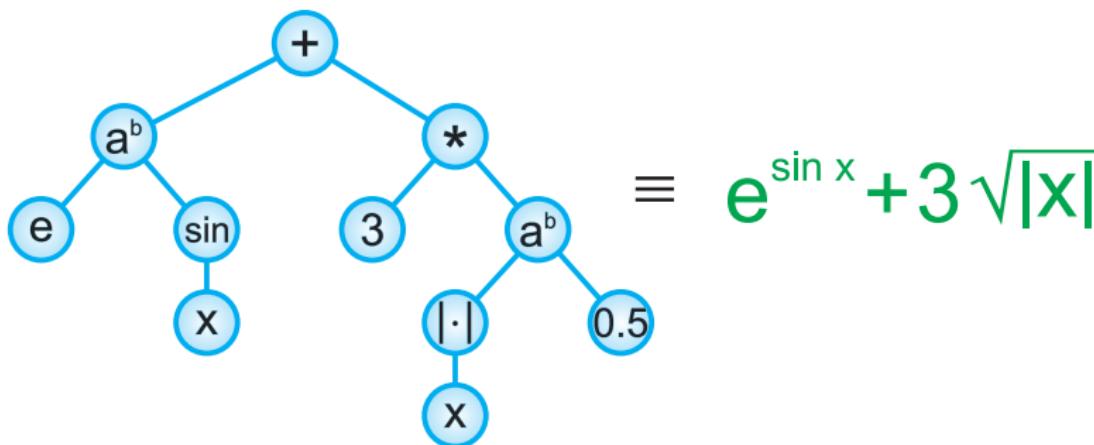
# Search Space

- How to encode such a heuristic function?
- Mathematical functions can well be represented as tree data structures



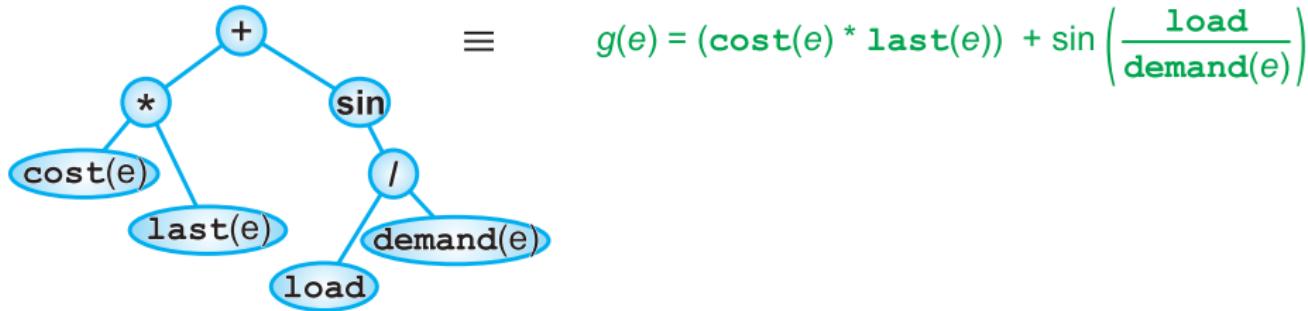
# Search Space

- How to encode such a heuristic function?
- Mathematical functions can well be represented as tree data structures and evolved with GP/Symbolic Regression [4, 15]



# Search Space

- How to encode such a heuristic function?
- Mathematical functions can well be represented as tree data structures and evolved with GP/Symbolic Regression [4, 15]
- Here: use GP to evolve the heuristic function



# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:

$a+b$

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:  
 $a+b$ ,  $\mathbf{a}-\mathbf{b}$

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:  
 $a+b$ ,  $a-b$ ,  $\mathbf{a} \star \mathbf{b}$

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:  
 $a+b$ ,  $a-b$ ,  $a*b$ ,  $a/b$

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:  
 $a+b, a-b, a*b, a/b, \max\{a, b\}$

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:  
 $a+b, a-b, a*b, a/b, \max\{a, b\}, \exp(a)$

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:  
 $a+b, a-b, a*b, a/b, \max\{a, b\}, \exp(a), \sin(a)$

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:  
 $a+b, a-b, a*b, a/b, \max\{a, b\}, \exp(a), \sin(a), \texttt{angle}(\mathbf{a}, \mathbf{b})$

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:  
 $a+b, a-b, a*b, a/b, \max\{a, b\}, \exp(a), \sin(a), \text{angle}(a, b)\dots$
- ... and some special terminals

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:  
 $a+b, a-b, a*b, a/b, \max\{a, b\}, \exp(a), \sin(a), \text{angle}(a, b)\dots$
- ... and some special terminals:
  - **demand( $e$ )**  
normalized demand  $\mathbf{R}(e)$  of edge

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:  
 $a+b, a-b, a*b, a/b, \max\{a, b\}, \exp(a), \sin(a), \text{angle}(a, b)\dots$
- ... and some special terminals:
  - **demand( $e$ )**
  - **load**  
normalized remaining amount of the product in vehicle

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:  
 $a+b, a-b, a*b, a/b, \max\{a, b\}, \exp(a), \sin(a), \text{angle}(a, b)\dots$
- ... and some special terminals:
  - **demand**( $e$ )
  - **load**
  - **cost**( $e$ )  
normalized cost of servicing edge  $e \in E$

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:  
 $a+b, a-b, a\star b, a/b, \max\{a, b\}, \exp(a), \sin(a), \text{angle}(a, b)\dots$
- ... and some special terminals:
  - **demand**( $e$ )
  - **load**
  - **cost**( $e$ )
    - normalized cost of servicing edge  $e \in E$
    - if necessary, includes costs for trip back to depot

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:  
 $a+b, a-b, a*b, a/b, \max\{a, b\}, \exp(a), \sin(a), \text{angle}(a, b)\dots$
- ... and some special terminals:
  - **demand**( $e$ )
  - **load**
  - **cost**( $e$ )
  - **depotCost**( $e$ )  
normalized costs to reach the depot from the *end* of  $e$

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:  
 $a+b, a-b, a*b, a/b, \max\{a, b\}, \exp(a), \sin(a), \text{angle}(a, b)\dots$
- ... and some special terminals:
  - **demand**( $e$ )
  - **load**
  - **cost**( $e$ )
  - **depotCost**( $e$ )
  - **satisfied**  
fraction of edges with non-zero demand that are already satisfied

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:  
 $a+b, a-b, a*b, a/b, \max\{a, b\}, \exp(a), \sin(a), \text{angle}(a, b)\dots$
- ... and some special terminals:
  - **demand**( $e$ )
  - **load**
  - **cost**( $e$ )
  - **depotCost**( $e$ )
  - **satisfied**
  - **last**( $e$ )

heuristic value assigned to the edge  $e$  in the *previous* selection round

# Search Space

- Functions  $g$  composed of basic (protected) mathematical operators:  
 $a+b, a-b, a*b, a/b, \max\{a, b\}, \exp(a), \sin(a), \text{angle}(a, b)\dots$
- ... and some special terminals:
  - **demand**( $e$ )
  - **load**
  - **cost**( $e$ )
  - **depotCost**( $e$ )
  - **satisfied**
  - **last**( $e$ )

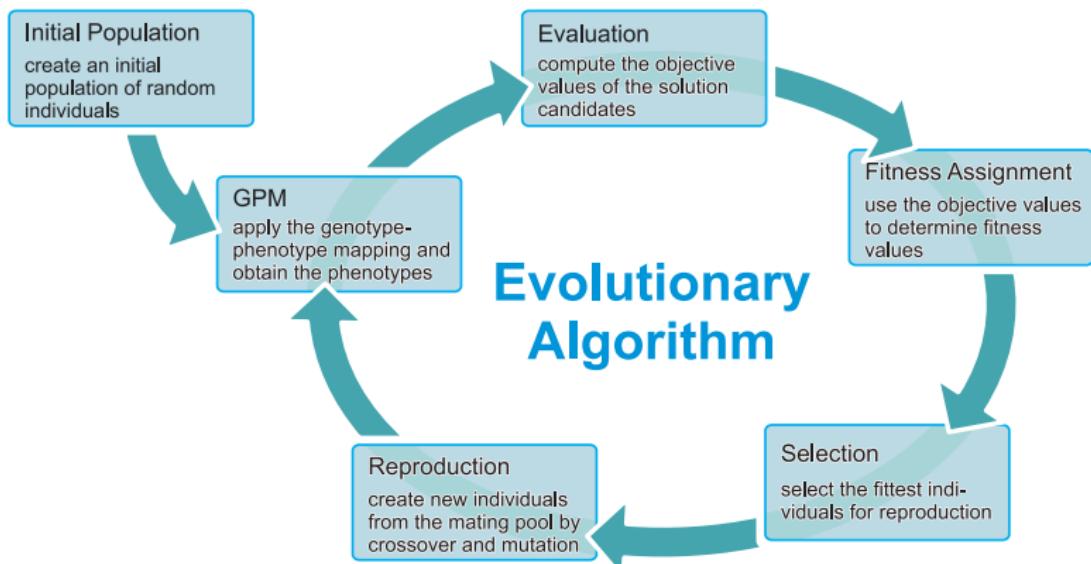
⇒ iterative building process with feedback from environment and previous steps

- 1 Introduction
- 2 CARP
- 3 Phenotypes & Objective
- 4 Developmental Approach
- 5 Experiments
- 6 Summary

# Experiments

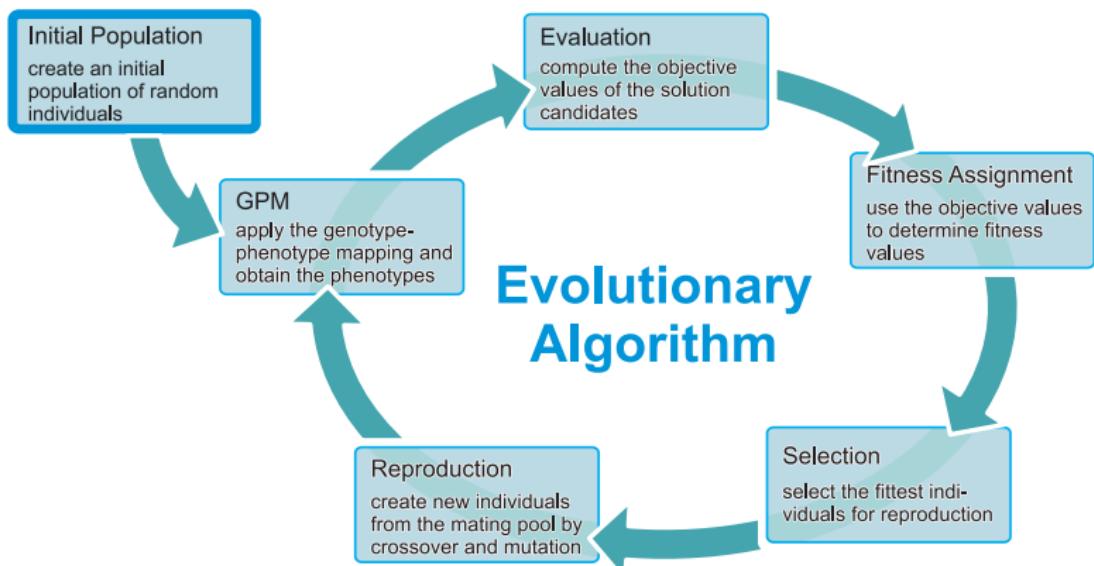
# Genetic Programming

- Optimization method: Genetic Programming<sup>[4–6]</sup>



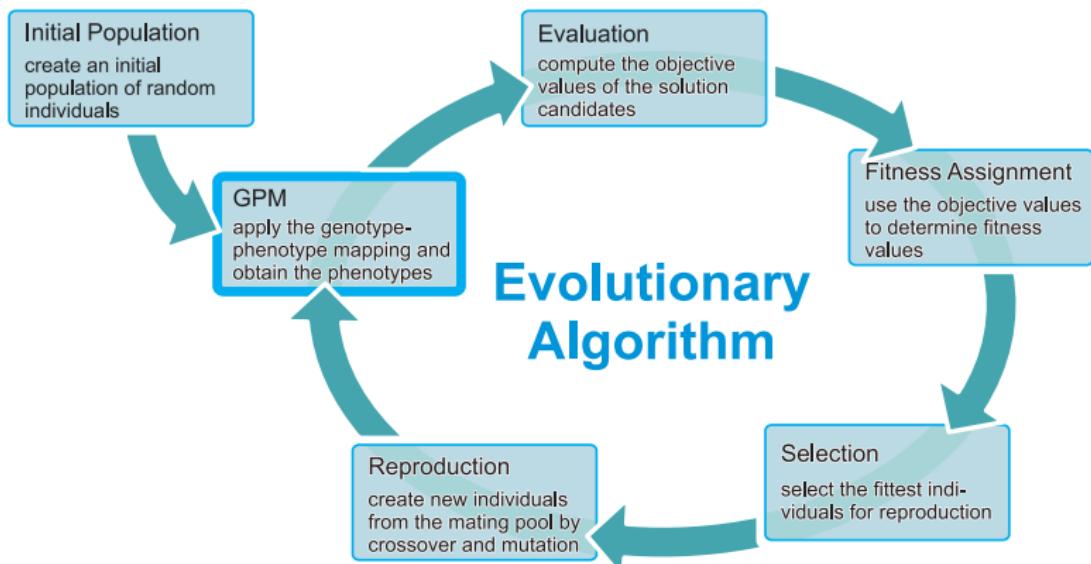
# Genetic Programming

- Ramped-half-and-half<sup>[4]</sup>



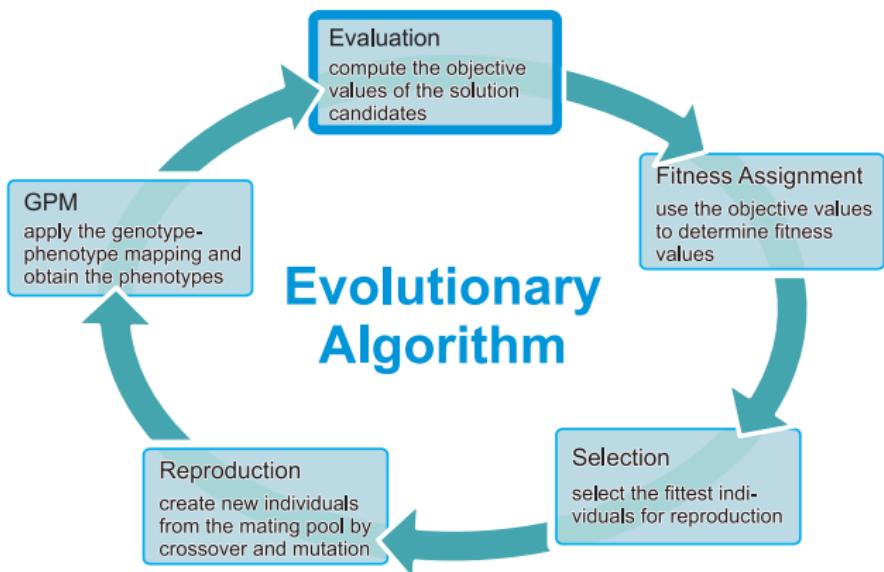
# Genetic Programming

- Developmental genotype-phenotype mapping: from heuristic function to schedule



# Genetic Programming

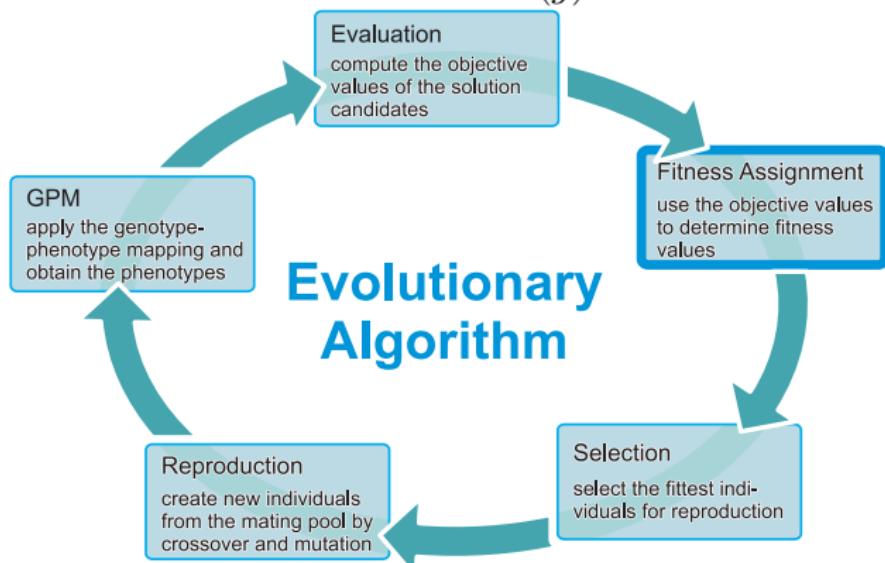
- Objective function  $f$ : total cost of plan  $x$



# Genetic Programming

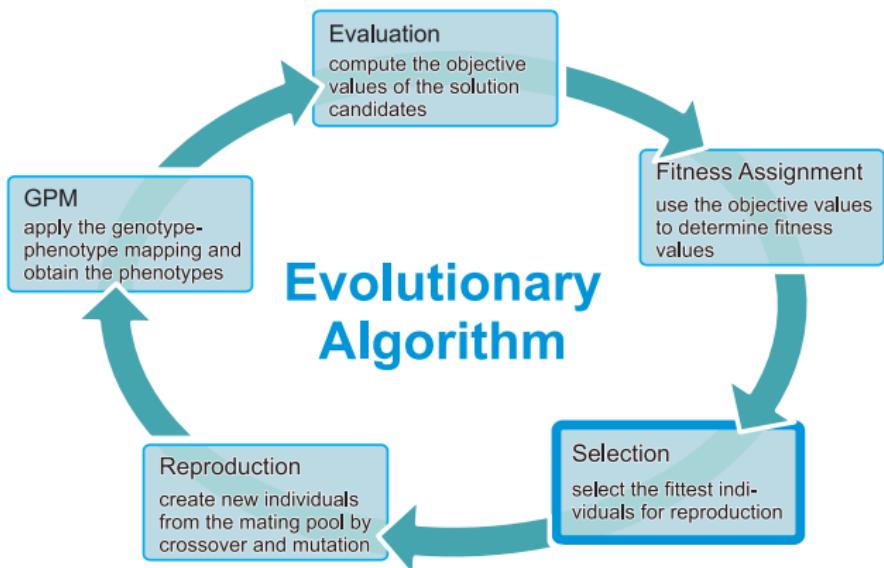
- Fitness: combine  $f$  with size of heuristic

$$v(g) = f(\text{gpm}(g)) - \frac{1}{\text{treeSize}(g)} \quad (1)$$



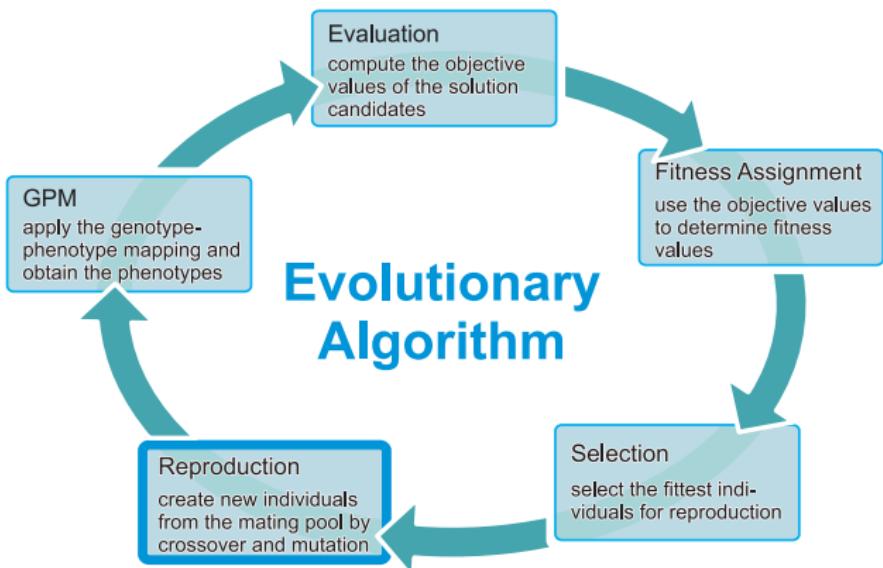
# Genetic Programming

- $(\mu + \lambda)$  population treatment (truncation selection,  $\mu = \lambda = 48$ )



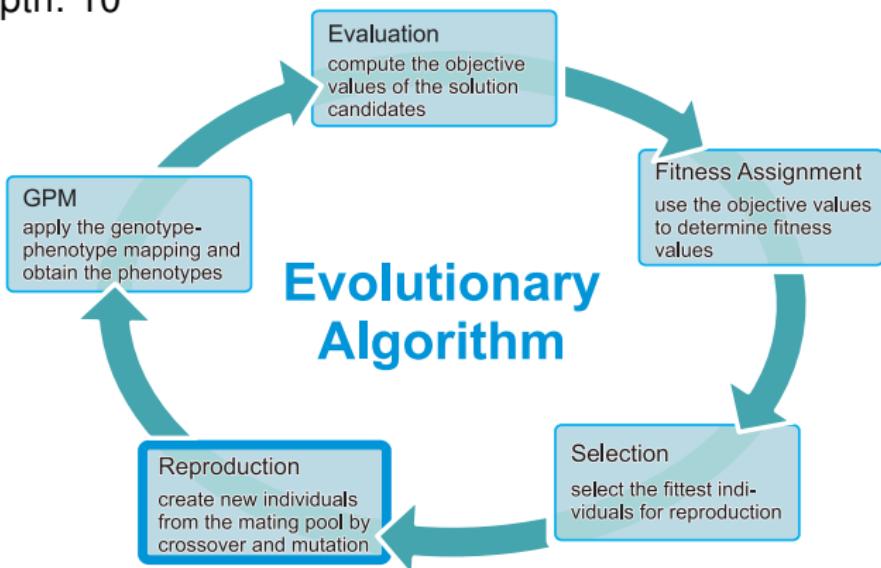
# Genetic Programming

- sub-tree exchange crossover, sub-tree replacement mutation, and mathematical simplification in proportion 2:5:2



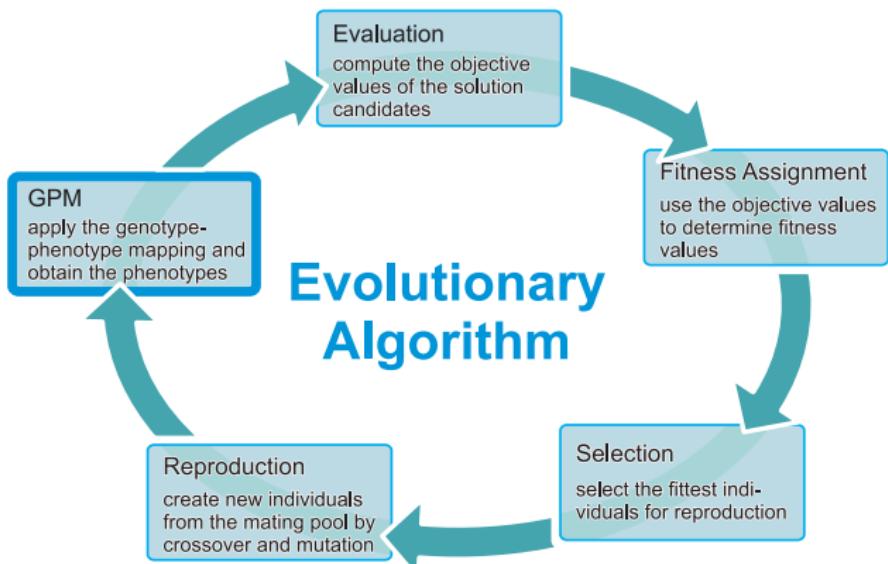
# Genetic Programming

- sub-tree exchange crossover, sub-tree replacement mutation, and mathematical simplification in proportion 2:5:2
- maximum tree depth: 10



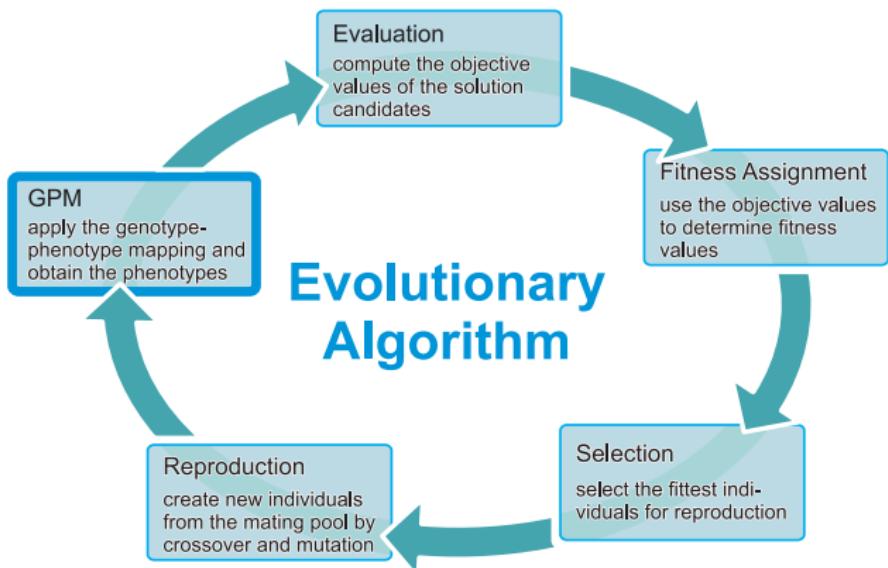
# Genetic Programming

- 16 384 function evaluations (FEs)



# Genetic Programming

- 16 384 function evaluations (FEs)
- Independent restart after 1536 FEs (32 generations) without improvements



# Benchmarks

- Experiments with 5 benchmark sets:
  - *gdb*<sup>[16]</sup>,
  - *val*<sup>[17]</sup>,
  - *egl*<sup>[18–20]</sup>,
  - *br-egl*<sup>[21, 22]</sup>, and
  - *kshs*<sup>[23]</sup>
- 30 runs for each benchmark

# Results

- Results are not as good as current best algorithms **RTS** [21] and  
**MAENS** [24]

# Results

- Results are not as good as current best algorithms **RTS** [21] and **MAENS** [24]:
  - often quite close

# Results

- Results are not as good as current best algorithms **RTS** [21] and **MAENS** [24]:
  - often quite close
  - but up to 12% off compared to RTS/MAENS

# Results

- Results are not as good as current best algorithms **RTS** [21] and **MAENS** [24]:
  - often quite close
  - but up to 12% off compared to **RTS/MAENS**
  - even more compared to lower bound...

# Results

- Results are not as good as current best algorithms **RTS** [21] and **MAENS** [24]:
  - often quite close
  - but up to 12% off compared to RTS/MAENS
  - even more compared to lower bound...
- Reasons are

# Results

- Results are not as good as current best algorithms **RTS** [21] and **MAENS** [24]:
  - often quite close
  - but up to 12% off compared to **RTS/MAENS**
  - even more compared to lower bound...
- Reasons are:
  - results obtained with **RTS/MAENS** are already global optima (or close to them)

# Results

- Results are not as good as current best algorithms **RTS** [21] and **MAENS** [24]:
  - often quite close
  - but up to 12% off compared to **RTS/MAENS**
  - even more compared to lower bound...
- Reasons are:
  - results obtained with **RTS/MAENS** are already global optima (or close to them)
  - indirect encoding only represents subset of possible solutions [13]

# Results

- Results are not as good as current best algorithms **RTS** [21] and **MAENS** [24]:
  - often quite close
  - but up to 12% off compared to **RTS/MAENS**
  - even more compared to lower bound...
- Reasons are:
  - results obtained with **RTS/MAENS** are already global optima (or close to them)
  - indirect encoding only represents subset of possible solutions [13]
  - e.g., **MAENS** used 90 000 FEs [25]

# Results

- Results are not as good as current best algorithms **RTS** [21] and **MAENS** [24]:
  - often quite close
  - but up to 12% off compared to **RTS/MAENS**
  - even more compared to lower bound...
- Reasons are:
  - results obtained with **RTS/MAENS** are already global optima (or close to them)
  - indirect encoding only represents subset of possible solutions [13]
  - e.g., **MAENS** used 90 000 FEs [25]
- So why even explore this method any further?

# Scalability?

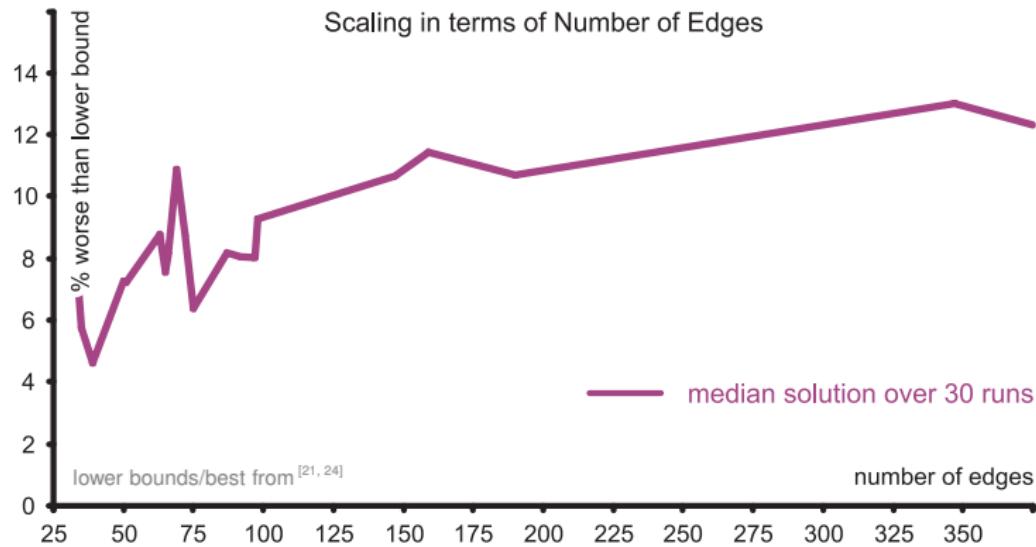
- Genotype size independent of problem instance size<sup>[13]</sup>

# Scalability?

- Genotype size independent of problem instance size<sup>[13]</sup>
- If problems get larger, results should not get too much worse...?<sup>[13]</sup>

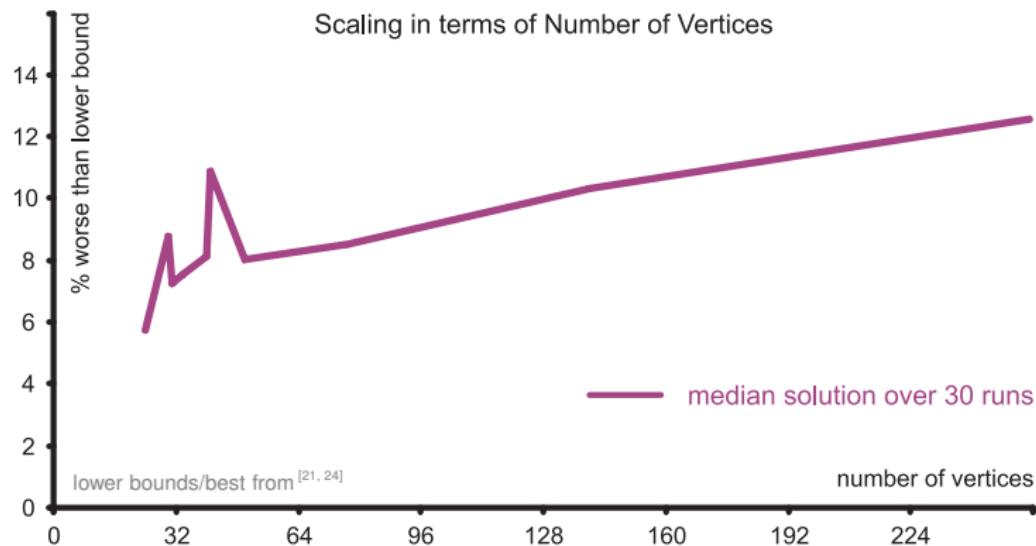
# Scalability?

- Genotype size independent of problem instance size [13]
- If problems get larger, results should not get too much worse...? [13]



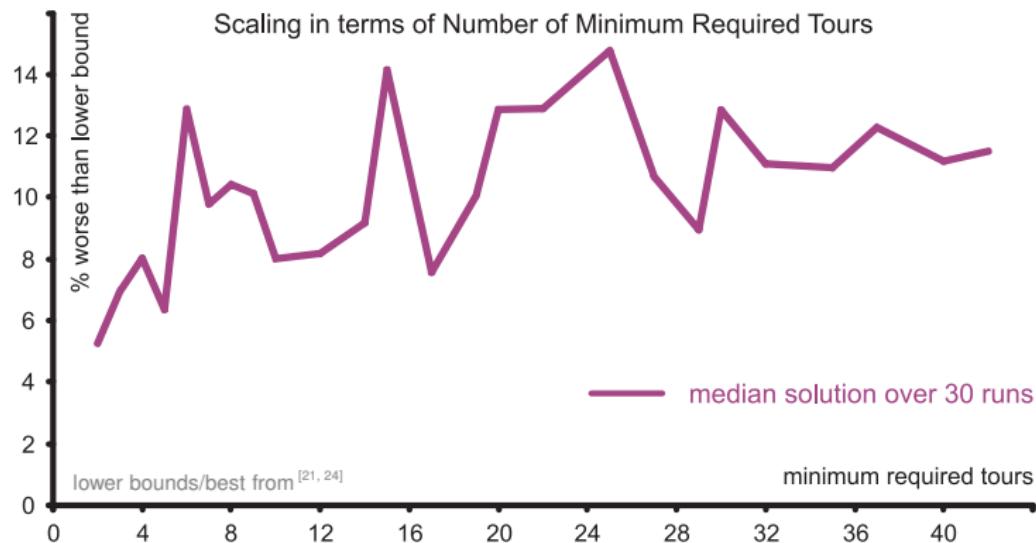
# Scalability?

- Genotype size independent of problem instance size [13]
- If problems get larger, results should not get too much worse...? [13]



# Scalability?

- Genotype size independent of problem instance size [13]
- If problems get larger, results **do** not get too much worse.



# Genotype Reuse?

- Can genotypes be used for seeding?

# Genotype Reuse?

- Can genotypes be used for seeding?
- Second set of experiments

# Genotype Reuse?

- Can genotypes be used for seeding?
- Second set of experiments:
  - setting  $\text{GP}^*$ : identical to original configuration  $\text{GP}$

# Genotype Reuse?

- Can genotypes be used for seeding?
- Second set of experiments:
  - setting  $\text{GP}^*$ : identical to original configuration  $\text{GP}$ , but
  - seeded with best solutions found on **some** of the benchmarks

# Genotype Reuse?

- Can genotypes be used for seeding?
- Second set of experiments:
  - setting  $\text{GP}^*$ : identical to original configuration  $\text{GP}$ , but
  - seeded with best solutions found on **some** of the benchmarks and
  - half as many FEs

# Genotype Reuse?

- Can genotypes be used for seeding?
- Second set of experiments:
  - setting  $\text{GP}^*$ : identical to original configuration  $\text{GP}$ , but
  - seeded with best solutions found on **some** of the benchmarks and
  - half as many FEs
- $\text{GP}^*$  is almost always better than  $\text{GP}$  although much fewer FEs

# Genotype Reuse?

- Genotypes **can** be used for seeding?
- Second set of experiments:
  - setting **GP<sup>\*</sup>**: identical to original configuration **GP**, but
  - seeded with best solutions found on **some** of the benchmarks and
  - half as many FEs
- **GP<sup>\*</sup>** is almost always better than **GP** although much fewer FEs
- Even on sets which are **unseen** (not used for seeding) *and* have **different scale** (*br-egl* and *kshs*)

# Genotype Reuse?

- Genotypes **can** be used for seeding?
- Second set of experiments:
  - setting **GP<sup>\*</sup>**: identical to original configuration **GP**, but
  - seeded with best solutions found on **some** of the benchmarks and
  - half as many FEs
- **GP<sup>\*</sup>** is almost always better than **GP** although much fewer FEs
- Even on sets which are **unseen** (not used for seeding) *and* have **different scale** (*br-egl* and *kshs*)
- Seeded runs with **GP<sup>\*</sup>** can solve problems **much faster** and with **much better quality** than **GP**!

# Dynamic Scenarios?

- Can genotypes be directly reused for other/modified/dynamic problem instances? ( $\Rightarrow$  Hyperheuristic<sup>[26, 27]</sup>)

# Dynamic Scenarios?

- Can genotypes be directly reused for other/modified/dynamic problem instances? ( $\Rightarrow$  Hyperheuristic<sup>[26, 27]</sup>)
- Third set of experiments

# Dynamic Scenarios?

- Can genotypes be directly reused for other/modified/dynamic problem instances? ( $\Rightarrow$  Hyperheuristic<sup>[26, 27]</sup>)
- Third set of experiments:
  - test successful genotypes  $g$  on modified scenarios

# Dynamic Scenarios?

- Can genotypes be directly reused for other/modified/dynamic problem instances? ( $\Rightarrow$  Hyperheuristic<sup>[26, 27]</sup>)
- Third set of experiments:
  - test successful genotypes  $g$  on modified scenarios and compare with 7 runs of **GP\***

# Dynamic Scenarios?

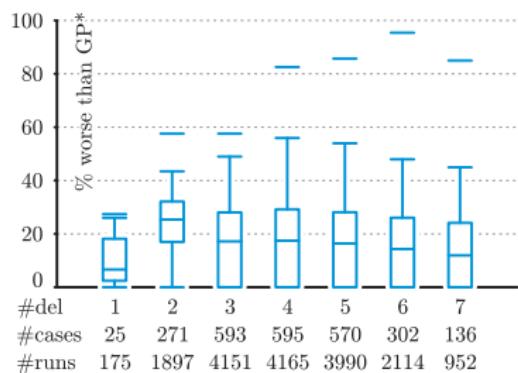
- Can genotypes be directly reused for other/modified/dynamic problem instances? ( $\Rightarrow$  Hyperheuristic<sup>[26, 27]</sup>)
- Third set of experiments:
  - test successful genotypes  $g$  on modified scenarios and compare with 7 runs of **GP\***
  - derive several new scenarios from the two cases *gdb10* (small scale) and *val7C* (larger scale)

# Dynamic Scenarios?

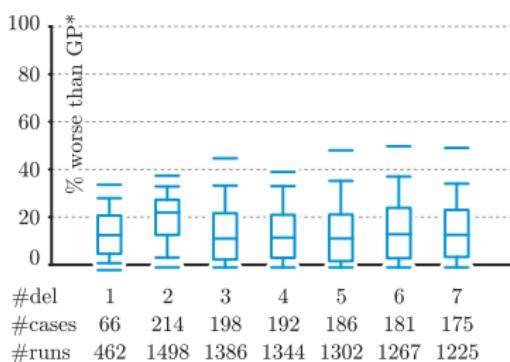
- Can genotypes be directly reused for other/modified/dynamic problem instances? ( $\Rightarrow$  Hyperheuristic<sup>[26, 27]</sup>)
- Third set of experiments:
  - test successful genotypes  $g$  on modified scenarios and compare with 7 runs of **GP\***
  - derive several new scenarios from the two cases *gdb10* (small scale) and *val7C* (larger scale)
  - with  $\#del$  edges removed (for  $\#del \in 1 \dots 7$ )

# Dynamic Scenarios?

- Can genotypes be directly reused for other/modified/dynamic problem instances? ( $\Rightarrow$  Hyperheuristic [26, 27])



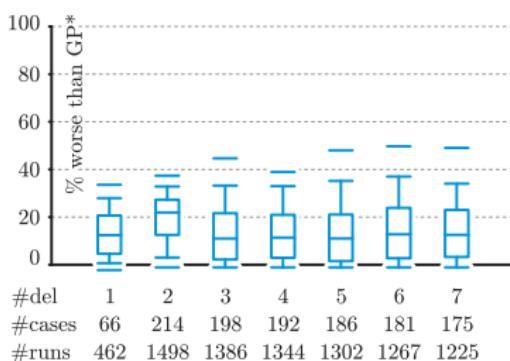
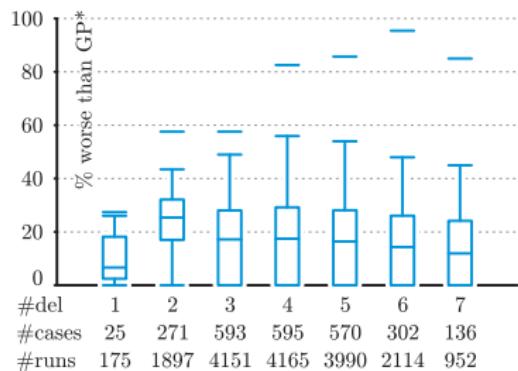
Box plot: static heuristic vs.  $GP^*$  in the *gdb10* scenarios.



Box plot: static heuristic vs.  $GP^*$  in the *val7C* scenarios.

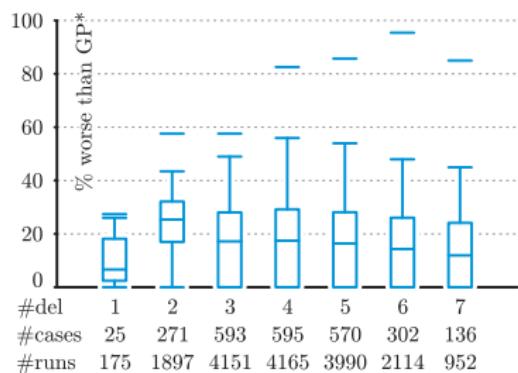
# Dynamic Scenarios?

- Can genotypes be directly reused for other/modified/dynamic problem instances? ( $\Rightarrow$  Hyperheuristic [26, 27])
  - similar performance changes in both scenarios

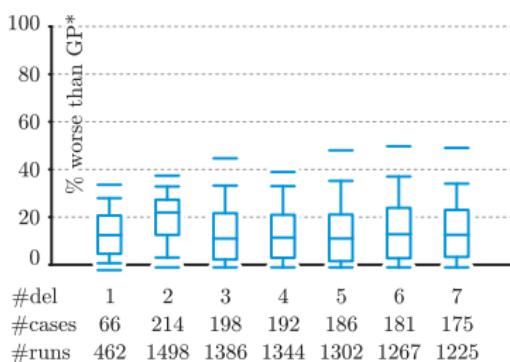


# Dynamic Scenarios?

- Can genotypes be directly reused for other/modified/dynamic problem instances? ( $\Rightarrow$  Hyperheuristic [26, 27])
  - similar performance changes in both scenarios
  - deleting edges in larger graph has lesser impact



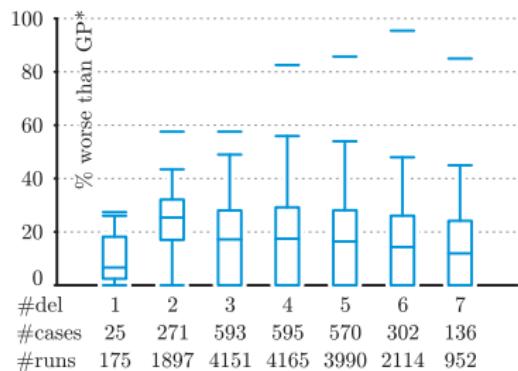
Box plot: static heuristic vs.  $GP^*$  in the *gdb10* scenarios.



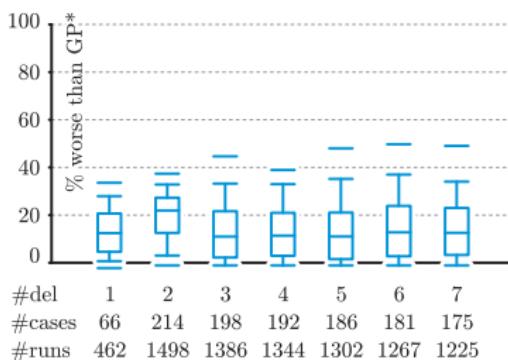
Box plot: static heuristic vs.  $GP^*$  in the *val7C* scenarios.

# Dynamic Scenarios?

- Genotypes can be directly reused for other/modified/dynamic problem instances. ( $\Rightarrow$  Hyperheuristic [26, 27])
  - 25% quantile often 0 or close to 0



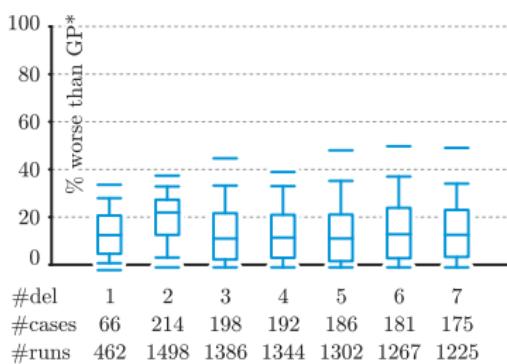
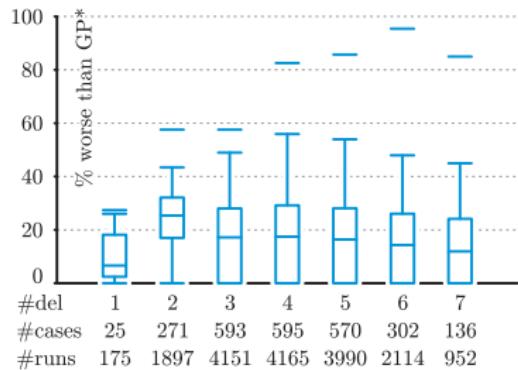
Box plot: static heuristic vs.  $GP^*$  in the *gdb10* scenarios.



Box plot: static heuristic vs.  $GP^*$  in the *val7C* scenarios.

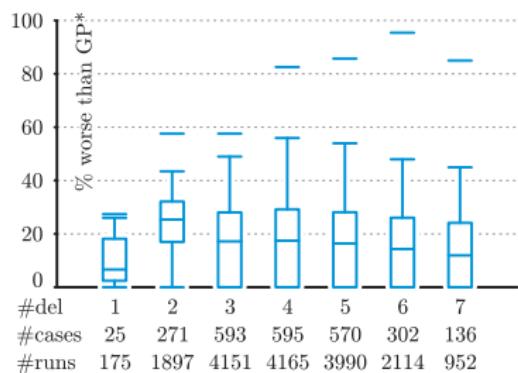
# Dynamic Scenarios?

- Genotypes can be directly reused for other/modified/dynamic problem instances if absolutely necessary.
  - 25% quantile often 0 or close to 0... but median often  $\geq 10\%$ ...

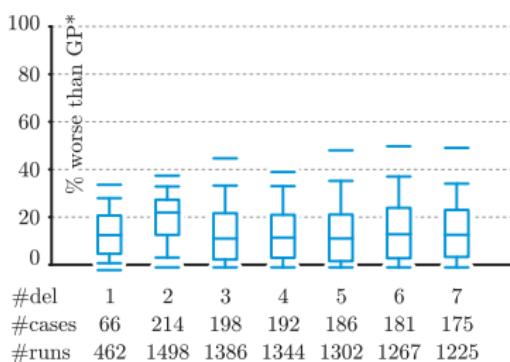


# Dynamic Scenarios?

- Genotypes can be directly reused for other/modified/dynamic problem instances if absolutely necessary.
  - 25% quantile often 0 or close to 0... but median often  $\geq 10\%$ ...
  - direct re-use of genotype fast, but not so good...



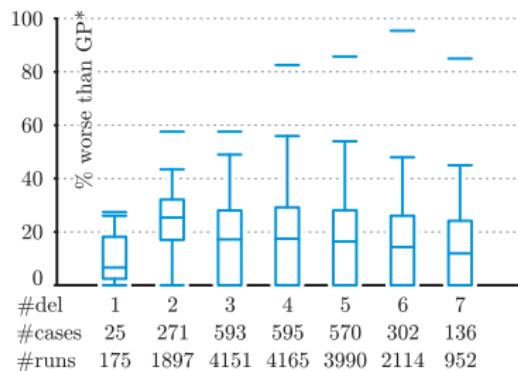
Box plot: static heuristic vs.  $GP^*$  in the *gdb10* scenarios.



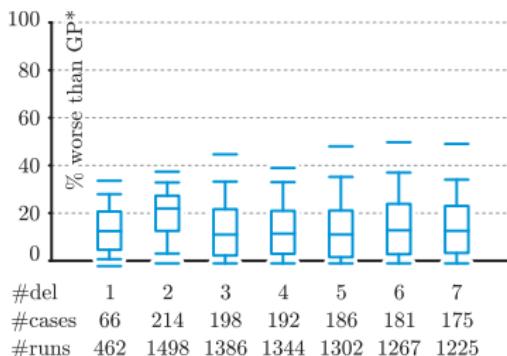
Box plot: static heuristic vs.  $GP^*$  in the *val7C* scenarios.

# Dynamic Scenarios?

- Genotypes can be directly reused for other/modified/dynamic problem instances if absolutely necessary.
  - 25% quantile often 0 or close to 0... but median often  $\geq 10\%$ ...
  - direct re-use of genotype fast, but not so good (overfitting, no Hyperheuristic)...



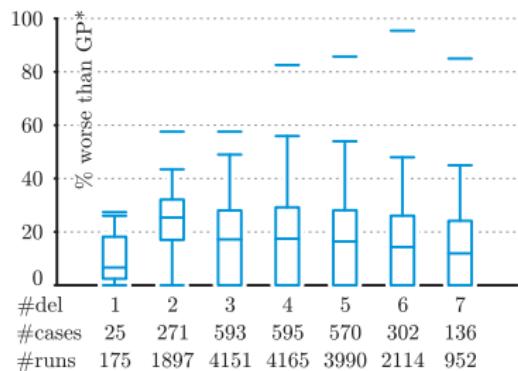
Box plot: static heuristic vs.  $GP^*$  in the *gdb10* scenarios.



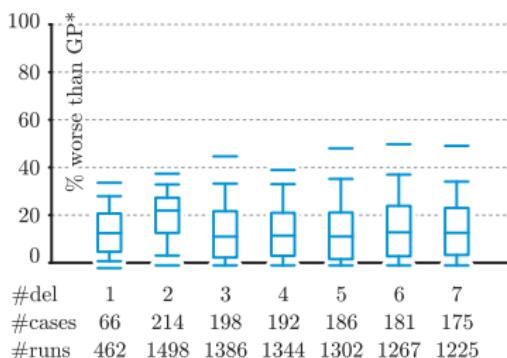
Box plot: static heuristic vs.  $GP^*$  in the *val7C* scenarios.

# Dynamic Scenarios?

- Genotypes can be directly reused, but better do a fast re-adjustment run with **GP\***.
  - 25% quantile often 0 or close to 0... but median often  $\geq 10\%$ ...
  - direct re-use of genotype fast, but not so good (overfitting, no Hyperheuristic)... better: a quick re-adjustment run with **GP\***



Box plot: static heuristic vs. **GP\*** in the *gdb10* scenarios.



Box plot: static heuristic vs. **GP\*** in the *val7C* scenarios.

- 1 Introduction
- 2 CARP
- 3 Phenotypes & Objective
- 4 Developmental Approach
- 5 Experiments
- 6 Summary

# Summary

# Summary

- Developmental representation for CARP

# Summary

- Developmental representation for CARP
- Several advantages

# Summary

- Developmental representation for CARP
- Several advantages:
  - genotype size independent of problem instance size

# Summary

- Developmental representation for CARP
- Several advantages:
  - genotype size independent of problem instance size
  - solution quality degenerates slowly with problem scale

# Summary

- Developmental representation for CARP
- Several advantages:
  - genotype size independent of problem instance size,
  - solution quality degenerates slowly with problem scale
  - genotypes can be re-used for seeding

# Summary

- Developmental representation for CARP
- Several advantages:
  - genotype size independent of problem instance size,
  - solution quality degenerates slowly with problem scale
  - genotypes can be re-used for seeding and
  - can build solutions in dynamic scenarios (though not very efficient)

# Summary

- Developmental representation for CARP
- Several advantages:
  - genotype size independent of problem instance size,
  - solution quality degenerates slowly with problem scale
  - genotypes can be re-used for seeding and
  - can build solutions in dynamic scenarios (though not very efficient)
- Disadvantages

# Summary

- Developmental representation for CARP
- Several advantages:
  - genotype size independent of problem instance size,
  - solution quality degenerates slowly with problem scale
  - genotypes can be re-used for seeding and
  - can build solutions in dynamic scenarios (though not very efficient)
- Disadvantages:
  - not (yet) as good as best algorithms

# Summary

- Developmental representation for CARP
- Several advantages:
  - genotype size independent of problem instance size,
  - solution quality degenerates slowly with problem scale
  - genotypes can be re-used for seeding and
  - can build solutions in dynamic scenarios (though not very efficient)
- Disadvantages:
  - not (yet) as good as best algorithms
  - GPM more complex than trivial mapping

# Summary

- Developmental representation for CARP
- Several advantages:
  - genotype size independent of problem instance size,
  - solution quality degenerates slowly with problem scale
  - genotypes can be re-used for seeding and
  - can build solutions in dynamic scenarios (though not very efficient)
- Disadvantages:
  - not (yet) as good as best algorithms
  - GPM more complex than trivial mapping
- Future/current work:
  - iterate whole GPM with information from previous iteration (done)

# Summary

- Developmental representation for CARP
- Several advantages:
  - genotype size independent of problem instance size,
  - solution quality degenerates slowly with problem scale
  - genotypes can be re-used for seeding and
  - can build solutions in dynamic scenarios (though not very efficient)
- Disadvantages:
  - not (yet) as good as best algorithms
  - GPM more complex than trivial mapping
- Future/current work:
  - iterate whole GPM with information from previous iteration (done)
  - replace GP with MLP<sup>[28]</sup>/CMA-ES<sup>[29-31]</sup> (done)

# Summary

- Developmental representation for CARP
- Several advantages:
  - genotype size independent of problem instance size,
  - solution quality degenerates slowly with problem scale
  - genotypes can be re-used for seeding and
  - can build solutions in dynamic scenarios (though not very efficient)
- Disadvantages:
  - not (yet) as good as best algorithms
  - GPM more complex than trivial mapping
- Future/current work:
  - iterate whole GPM with information from previous iteration (done)
  - replace GP with MLP<sup>[28]</sup>/CMA-ES<sup>[29-31]</sup> (done)
  - extend method to other problems, e.g., TSP (done)

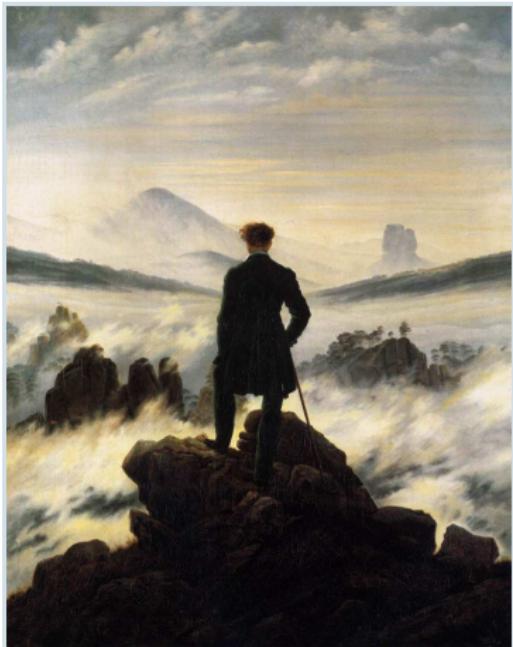
# Summary

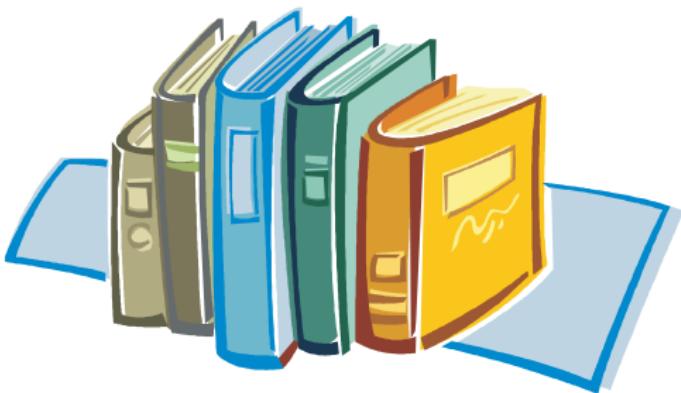
- Developmental representation for CARP
- Several advantages:
  - genotype size independent of problem instance size,
  - solution quality degenerates slowly with problem scale
  - genotypes can be re-used for seeding and
  - can build solutions in dynamic scenarios (though not very efficient)
- Disadvantages:
  - not (yet) as good as best algorithms
  - GPM more complex than trivial mapping
- Future/current work:
  - iterate whole GPM with information from previous iteration (done)
  - replace GP with MLP<sup>[28]</sup>/CMA-ES<sup>[29-31]</sup> (done)
  - extend method to other problems, e.g., TSP (done)
  - investigate other ways to utilize developmental mappings for logistic planning (in progress)

# 谢谢你们！

Thank you very much for  
your kind attention.

Any questions?





# References

# References I

1. Thomas Weise, Ké Táng, and Alexandre Devert. A Developmental Solution to (Dynamic) Capacitated Arc Routing Problems using Genetic Programming. In Terence Soule and Jason H. Moore, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'12)*, pages 831–838. Association for Computing Machinery (ACM): New York, NY, USA, 2012. 10.1145/2330163.2330278.
2. Thomas Weise. Representations for Logistic Planning, 2012.
3. Moshe Dror, editor. *Arc Routing: Theory, Solutions and Applications*. Springer-Verlag GmbH: Berlin, Germany, 2000. ISBN 0-7923-7898-9.
4. John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Bradford Books. MIT Press: Cambridge, MA, USA, December 1992. ISBN 0-262-11170-5. 1992 first edition, 1993 second edition.
5. Thomas Weise. *Global Optimization Algorithms – Theory and Application*. it-weise.de (self-published): Germany, 2009. URL <http://www.it-weise.de/projects/book.pdf>.
6. Riccardo Poli, William B. Langdon, and Nicholas Freitag McPhee. *A Field Guide to Genetic Programming*. Lulu Enterprises UK Ltd: London, UK, March 2008. ISBN 1-4092-0073-6. URL <http://www.gp-field-guide.org.uk/>. With contributions by John R. Koza.
7. Hisashi Handa, Dan Lin, Lee Chapman, and Xin Yao. Robust Solution of Salting Route Optimisation Using Evolutionary Algorithms. In Gary G. Yen, Simon M. Lucas, Gary B. Fogel, Graham Kendall, Ralf Salomon, Byoung-Tak Zhang, Carlos Artemio Coello Coello, and Thomas Philip Runarsson, editors, *Proceedings of the IEEE Congress on Evolutionary Computation (CEC'06), 2006 IEEE World Congress on Computation Intelligence (WCCI'06)*, pages 3098–3105. IEEE Computer Society: Piscataway, NJ, USA, IEEE Computer Society: Piscataway, NJ, USA, 2006. 10.1109/CEC.2006.1688701. URL [http://escholarship.lib.okayama-u.ac.jp/industrial\\_engineering/2/](http://escholarship.lib.okayama-u.ac.jp/industrial_engineering/2/).
8. Hisashi Handa, Lee Chapman, and Xin Yao. Robust Route Optimization for Gritting/Salting Trucks: A CERCIA Experience. *IEEE Computational Intelligence Magazine*, 1(1):6–9, February 2006. 10.1109/MCI.2006.1597056. URL <http://www.cs.bham.ac.uk/~xin/papers/SaltingCIMagazine.pdf>.
9. Méigǔ Guǎn. Graphic Programming Using Odd or Even Points. *Chinese Mathematics*, 1:273–277, 1962.

# References II

10. Horst A. Eiselt, Michael Gendreau, and Gilbert Laporte. Arc Routing Problems, Part I: The Chinese Postman Problem. *Operations Research*, 43(2):231–242, March–April 1995.
11. MS2012BMPPU. Bing Maps: Philadelphia, PA, USA, July 1, 2012. URL <http://cn.bing.com/maps/#Y3A9MzkuOTQ4NzYxMDk4ODg3M34tNzUuMTYzMjc5MDI2NzQ2NzUmbHZsPTE2JnN0eT1y>.
12. Robert W Floyd. Algorithm 97 (SHORTEST PATH). *Communications of the ACM (CACM)*, 5(6):345, June 1, 1962. 10.1145/367766.368168.
13. Alexandre Devert, Thomas Weise, and Kē Táng. A Study on Scalable Representations for Evolutionary Optimization of Ground Structures. *Evolutionary Computation*, 20, 2012. 10.1162/EVCO\_a\_00054. URL <http://www.marmakoide.org/download/publications/devweita-ecj-preprint.pdf>.
14. Alexandre Devert. *Building Processes Optimization: Toward an Artificial Ontogeny based Approach*. PhD thesis, Université Paris-Sud, Ecole Doctorale d'Informatique: Paris, France and Institut National de Recherche en Informatique et en Automatique (INRIA), Centre de Recherche Saclay – Île-de-France: Orsay, France, May 2009.
15. Jin Li, Xīn Yáo, Colin Frayn, Habib G. Khosroshahi, and Somak Raychaudhury. An Evolutionary Approach to Modeling Radial Brightness Distributions in Elliptical Galaxies. In Xīn Yáo, Edmund K. Burke, José Antonio Lozano, Jim Smith, Juan Julián Merelo-Guervós, John A. Bullinaria, Jonathan E. Rowe, Peter Tiño, Ata Kabán, and Hans-Paul Schwefel, editors, *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, volume 3242/2004 of *Lecture Notes in Computer Science (LNCS)*, pages 591–601. Springer-Verlag GmbH: Berlin, Germany, 2008. 10.1007/978-3-540-30217-9\_60. URL [http://www.cs.bham.ac.uk/~xin/papers/Li\\_ppsn314.pdf](http://www.cs.bham.ac.uk/~xin/papers/Li_ppsn314.pdf).
16. James Stone DeArmon. A Comparison of Heuristics for the Capacitated Chinese Postman Problem. Master's thesis, University of Maryland: College Park, MD, USA, 1981.
17. Enrique Benavent, V. Campos, A. Corberán, and E. Mota. The Capacitated Arc Routing Problem. Lower Bounds. *Networks*, 22(7):669–690, December 1992. 10.1002/net.3230220706.
18. Richard W. Eglese. Routing Winter Gritting Vehicles. *Discrete Applied Mathematics – The Journal of Combinatorial Algorithms, Informatics and Computational Sciences*, 48(3):231–244, February 15, 1994. 10.1016/0166-218X(92)00003-5.

# References III

19. Richard W. Eglese and Leon Y. O. Li. A Tabu Search based Heuristic for Arc Routing with a Capacity Constraint and Time Deadline. In Ibrahim H. Osman and James Patrick Kelly, editors, *Meta-heuristics Theory and Applications*, pages 633–649. Kluwer Academic Publishers: Norwell, MA, USA, 1996.
20. Leon Y. O. Li and Richard W. Eglese. An Interactive Algorithm for Vehicle Routeing for Winter - Gritting. *The Journal of the Operational Research Society (JORS)*, 47(2):217–228, February 1996.
21. Yī Méi, Kē Táng, and Xīn Yáo. A Global Repair Operator for Capacitated Arc Routing Problem. *IEEE Transactions on Systems, Man, and Cybernetics – Part B: Cybernetics*, 39(3):723–734, June 2009. 10.1109/TSMCB.2008.2008906. URL [http://www.cs.bham.ac.uk/~xin/papers/TSMC09\\_MeiTangYao.pdf](http://www.cs.bham.ac.uk/~xin/papers/TSMC09_MeiTangYao.pdf).
22. José Brandão and Richard W. Eglese. A Deterministic Tabu Search Algorithm for the Capacitated Arc Routing Problem. *Computers & Operations Research*, 35(4):1112–1126, April 2008. 10.1016/j.cor.2006.07.007. URL <http://eprints.lancs.ac.uk/48773/1/Document.pdf>. Also: Lancaster University Management School Working Paper 2005/027.
23. M. Kiuchi, Y. Shinano, R. Hirabayashi, and Y. Saruwatari. An Exact Algorithm for the Capacitated Arc Routing Problem using Parallel Branch and Bound Method. In *Abstracts of the National Conference of the Operations Research Society of Japan*, pages 28–29. Operations Research Society of Japan (OSRJ): Tōkyō, Japan, 1995. Written in Japanese.
24. Kē Táng, Yī Méi, and Xīn Yáo. Memetic Algorithm with Extended Neighborhood Search for Capacitated Arc Routing Problems. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 13(5):1151–1166, October 2009. 10.1109/TEVC.2009.2023449. URL [http://staff.ustc.edu.cn/~ketang/papers/TangMeiYao\\_TEVC09.pdf](http://staff.ustc.edu.cn/~ketang/papers/TangMeiYao_TEVC09.pdf).
25. Yī Méi, Kē Táng, and Xīn Yáo. Decomposition-Based Memetic Algorithm for Multi-Objective Capacitated Arc Routing Problem. *IEEE Transactions on Evolutionary Computation (IEEE-EC)*, 15(2):151–165, April 2011. 10.1109/TEVC.2010.2051446. URL [http://ieee-cis.org/\\_files/EAC\\_Research\\_2009\\_Report\\_Mei.pdf](http://ieee-cis.org/_files/EAC_Research_2009_Report_Mei.pdf).
26. Edmund K. Burke, Matthew Hyde, Graham Kendall, Gabriela Ochoa, Ender Ozcan, and Rong Qu. A Survey of Hyper-Heuristics. Computer Science Technical Report NOTTCS-TR-SUB-0906241418-2747, University of Nottingham, School of Computer Science & Information Technology: Nottingham, UK, March 2009. URL <http://www.cs.nott.ac.uk/TR/SUB/SUB-0906241418-2747.pdf>.

# References IV

27. Peter Ross. Hyper-Heuristics. In Edmund K. Burke and Graham Kendall, editors, *Search Methodologies – Introductory Tutorials in Optimization and Decision Support Techniques*, chapter 17, pages 529–556. Springer Science+Business Media, Inc.: New York, NY, USA, 2005. 10.1007/0-387-28356-0\_17.
28. Simon S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall International Inc.: Upper Saddle River, NJ, USA, 2nd, illustrated edition, 1999. ISBN 0-13-273350-1.
29. Nikolaus Hansen and Andreas Ostermeier. Completely Derandomized Self-Adaptation in Evolution Strategies. *Evolutionary Computation*, 9(2):159–195, 2001. URL <http://www.bionik.tu-berlin.de/user/niko/cmaartic.pdf>.
30. Nikolaus Hansen and Stefan Kern. Evaluating the CMA Evolution Strategy on Multimodal Test Functions. In Xīn Yáo, Edmund K. Burke, José Antonio Lozano, Jim Smith, Juan Julián Merelo-Guervós, John A. Bullinaria, Jonathan E. Rowe, Peter Tiño, Ata Kabán, and Hans-Paul Schwefel, editors, *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature (PPSN VIII)*, volume 3242/2004 of *Lecture Notes in Computer Science (LNCS)*, pages 282–291. Springer-Verlag GmbH: Berlin, Germany, 2008. 10.1007/978-3-540-30217-9\_29.
31. Raymond Ros and Nikolaus Hansen. A Simple Modification in CMA-ES Achieving Linear Time and Space Complexity. In *Proceedings of 10th International Conference on Parallel Problem Solving from Nature (PPSN X)*, volume 5199/2008 of *Theoretical Computer Science and General Issues (SL 1)*, *Lecture Notes in Computer Science (LNCS)*, pages 296–305. Springer-Verlag GmbH: Berlin, Germany, 2008. 10.1007/978-3-540-87700-4\_30. URL <http://hal.inria.fr/docs/00/27/32/71/PS/RR-6498.ps>.